

Visual Analytics of Large Weighted Directed Graphs and Two-Dimensional Time-Dependent Data



vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Dipl.-Math. Tatiana Landesberger von Antburg,
geboren Tekušová

geboren in Bratislava, Slowakische Republik

Referenten der Arbeit: Prof. Dr. techn. Dieter W. Fellner
Technische Universität Darmstadt
Prof. dr. ir. Jarke J. van Wijk
Technische Universiteit Eindhoven, Niederlande

Tag der Einreichung: 17.5.2010
Tag der mündlichen Prüfung: 28.6.2010

Darmstädter Dissertation
D 17
Darmstadt, 2010

Acknowledgments

I would like to express my gratefulness to those people who contributed in various ways to the development of this thesis.

I would like to especially thank Prof. Dr. techn. Dieter W. Fellner for his continuous support in this endeavor. His inspiring and constructive comments during various stages of the process and the possibility to work on the thesis in the encouraging environment of Technische Universität Darmstadt were the basis for the completion of the project. I am thankful to Prof. dr. ir. Jarke J. van Wijk for acting as a second advisor. His useful comments on the thesis and on the state-of-the-art-report on visual analysis of graphs are much appreciated.

The thesis has been conducted at Fraunhofer IGD and in the Graphics Interactive Systems Group (GRIS) at Technische Universität Darmstadt. The head of Visual Search and Analysis Group Dr. Tobias Schreck has actively supported my scientific work during the whole thesis preparation phase and Priv. Doz. Dr. habil. Arjan Kuijper has contributed with useful comments to the draft in its final phase. I am thankful to Dr. Jörn Kohlhammer and Prof. Dr.-Ing. José L. Encarnação for founding the Visual Analytics topic at Fraunhofer IGD and introducing me to this interesting research area. During my work at Fraunhofer IGD and TU Darmstadt, I have collaborated with many colleagues who in this way contributed to this thesis. In particular, I am thankful to Dr. Christoph Hornung and Dr. Bettina Hornung for their numerous constructive suggestions on my current research and various related issues. I am grateful to Sebastian Bremm, Matthias Kirschner and Torsten Techmann for their active role in discussions on daily work topics and to Priv. Doc. Galina Paramei and her colleagues for the collaboration on perception studies. Furthermore, during my project work and supervising activities, I have worked with many students on the topics related to the thesis. These activities brought new inspirations to the work. In particular, I appreciated the collaboration with Robert Rehner, Melanie Görner and Jürgen Bernard and their longer-term engagement in the related projects.

Finally, I am very grateful to my family for their support in hard situations and in particular to my husband Julian for all his patience, countless helpful hints and encouraging words.

Abstract

Analysts need to effectively assess large amounts of data. Often, their focus is on two types of data: weighted directed graphs and two-dimensional time dependent data. These types of data are commonly examined in various application areas such as transportation, finance, or biology. The key elements in supporting the analysis are systems that seamlessly integrate interactive visualization techniques and data processing. The systems also need to offer the analyst the possibility to flexibly steer the analytical process.

In this thesis, we present new techniques providing such flexible integrated combinations with tight user involvement in the analytical process for the two selected data types.

We first develop new techniques for visual analysis of weighted directed graphs.

- We enhance the analysis of entity relationships by integration of algorithmic analysis of connections in interactive visualization.
- We improve the analysis of graph structure by several ways of motif-based analysis.
- We introduce interactive visual clustering of graph connected components for gaining overview of the data space.

Second, we develop new methods for visual analysis of two-dimensional time dependent data. We thereby combine animation and trajectory-based interactive visualizations with user-driven feature-based data analysis.

- We extend guidelines for the use of animation by conducting a perception study of motion direction change.
- We introduce interactive monitoring of a new set of data features in order to analyze the data dynamics.
- We present visual clustering of trajectories of individual entities using self-organizing maps (SOM) with user control of the clustering process.

As a basis for the development of the new approaches, we discuss the methodology of Visual Analytics and its related fields. We thereby extend classification of Information Visualization and Interaction techniques used in Visual Analytics systems.

The developed techniques can be used in various application domains such as finance and economics, geography, social science, biology, transportation, or meteorology. In the financial domain, the techniques support analysts in making investment decisions, in assessment of company value, or in analysis of economy structure. We demonstrate our new methods on two real world data sets: shareholder networks and time-varying risk-return data.

Zusammenfassung

Die Analyse großer Datenmengen ist in vielen Anwendungsgebieten eine wichtige Aufgabe. Dazu zählen zum einen die Biologie, Pharmazie und Verkehrsplanung als auch Sozial- und Wirtschaftswissenschaften, um nur einige Beispiele zu nennen. Diese Gebiete sind auf eine effektive und schnelle Analyse angewiesen, um zeitnah Entscheidungen treffen zu können. Insbesondere die Analyse internationaler Finanzmärkte rückt zunehmend in den Vordergrund. Finanzdatenanbieter stellen eine Vielzahl an Datensätzen aus unterschiedlichen Quellen bereit, beispielsweise Aktienkurse aus dem elektronischen Handel in Echtzeit. Diese großen Datenmengen müssen von Analysten effektiv ausgewertet werden, um eine schnelle und zugleich angemessene Reaktion auf Marktentwicklungen zu gewährleisten.

Der Fokus der Untersuchung ist oft auf zwei wichtige Datentypen gerichtet: Auf gewichtete, gerichtete Graphen und auf zweidimensionale, zeitabhängige Daten. Die wesentlichen Elemente, die diese Analyse unterstützen, sind Systeme, die nahtlos interaktive Visualisierungstechniken und Datenverarbeitung verbinden. Diese Systeme sollen dem Analysten die Möglichkeit bieten, den analytischen Prozess flexibel zu steuern.

In dieser Arbeit präsentieren wir neue Techniken, die diese flexibel integrierten Kombinationen mit enger Einbeziehung des Nutzers in den analytischen Prozess für die beiden ausgewählten Datentypen unterstützen.

Für die **visuelle Analyse gerichteter, gewichteter Graphen** wurden auf den Datentyp und ausgewählte Nutzungsszenarien spezialisierte Techniken entwickelt. Ziel der Graphanalyse ist es, Wissen über globale und lokale Strukturen des Graphen und damit über die Zusammenhänge zwischen den repräsentierten Entitäten zu erlangen. Dabei können verschiedenen Teilgebiete definiert werden. Erstens, die Analyse direkter Beziehungen von Entitäten untereinander. Zweitens, die Identifikation interessanter, etwa häufig auftretender Beziehungsmuster und drittens, der Vergleich verschiedener Graphen. Bei der reinen Visualisierung von Graphen liegt der Schwerpunkt auf der Entwicklung effizienter Layout- und Navigationstechniken. Um allerdings große Datenmengen effektiv untersuchen zu können, ist die enge Integration von Algorithmen zur Graphanalyse in der Graphenvisualisierung notwendig. Diese mächtige Kombination interaktiv nutzen zu können, ist ein Kernelement für eine erfolgreiche Analyse.

Der *Beitrag* dieser Arbeit zur visuellen, interaktiven Graphanalyse konzentriert sich auf die drei genannten Einsatzgebiete und beinhaltet:

1. Verbesserung der *Analyse der Beziehung zwischen Graphknoten*. Durch die interaktive Integration graphalgoritmischer Analyse- und Visualisierungsmethoden wird eine einfache und effektive Untersuchung der Zusammenhänge gewährleistet. Die Ergebnisse wurden veröffentlicht in [TK08].
2. Erweiterung der *visuellen, interaktiven Analyse von Graphmotiven*, die vordefinierte und interaktiv durch den Benutzer definierte Strukturen untersucht. Auf Basis dieser Motivdaten kann eine hierarchische Aggregation der Daten erfolgen, um verschiedene Abstraktionsebenen zu erzeugen. Dadurch können verschiedene Fragestellungen zur lokalen und globalen Graphenstruktur behandelt werden. Weiterhin wird die Motivanalyse zur Auswertung struktureller Graphänderungen (z.B. benutzerdefinierte "was-wäre-wenn-Szenarien) eingeführt. Wir haben dazu ein flexibles, visuelles System, das eine interaktive Kombination dieser Techniken unterstützt, vorgestellt [vLGRS09].

-
3. *Analyse und Vergleich mehrerer Graphen (mittels Clustering)* mit einem Schwerpunkt auf der Untersuchung vieler zusammenhängender Graphkomponenten. In der Untersuchung spielen strukturelle Ähnlichkeiten der Graphen eine entscheidende Rolle. Wir haben eine Technik entwickelt, die verschiedene strukturelle Eigenschaften berücksichtigt und gemäß derer die Graphen nach ihrer Ähnlichkeit gruppiert werden. Eine interaktive, visuelle Analyse bietet verschiedene Visualisierungen für unterschiedliche Aspekte wie die Exploration der Rohdaten, eine interaktive Auswahl der Clustering-Parameter und die Auswertung der Ergebnisse einschließlich ihrer Qualität. Die Ergebnisse wurden in [vLGS09] präsentiert.

Auf dem Gebiet der **visuellen Analyse zweidimensionaler, zeitabhängiger Daten** umfasst unser Beitrag die Entwicklung von Techniken zur Kombination von animations- oder trajektorienbasierter Visualisierung mit interaktiver feature-basierter Analyse.

Bei der Analyse zweidimensionaler, zeitabhängiger Daten sind verschiedene Eigenschaften der Daten zu berücksichtigen, beispielsweise die Struktur und Verteilung der Daten (Identifikation von Clustern und Ausreißern) oder die räumlichen Beziehungen zwischen Datenelementen. Diese Kriterien müssen nicht nur unabhängig zu jedem Zeitpunkt berücksichtigt werden, sondern auch ihre Veränderung über die Zeit ist ein wichtiger Faktor im Analyseprozess. Der Fokus der Analyse liegt auf der Untersuchung der Dynamik der einzelnen Datenpunkte, der Analyse der Dynamik von Gruppen von Datenelementen (und Punkten innerhalb von Gruppen), sowie auf der Analyse von Bewegungsmustern der Datenelemente.

In Bezug auf zwei-dimensionale Zeitreihen ist es notwendig, mögliche interaktive Visualisierungstechniken und die ihnen zugrunde liegenden Wahrnehmungsmuster zu untersuchen. Die meisten Methoden zur Untersuchung zweidimensionaler, zeitabhängiger Daten stammen aus dem Gebiet der Geovisualisierung und umfassen daher vor Allem auf dieses Anwendungsgebiet spezialisierte Techniken. Weiterhin beschäftigen sich viele Techniken entweder mit der Analyse zweidimensionaler zeitunabhängiger oder eindimensionaler zeitabhängiger Daten. Die Kombination aus beidem weist dabei interessante und relevante Fragestellungen auf.

In unserer Arbeit stellen wir folgende *Beiträge* zur visuellen Analyse zweidimensionaler, zeitabhängiger Daten vor:

1. Animation ist ein häufiges Mittel zur Visualisierung von zweidimensionaler, zeitabhängiger Daten. Dabei können verschiedene visuelle Attribute eingesetzt werden. Deren *Wirkung auf die menschliche Wahrnehmung* haben wir im Rahmen einer Benutzerstudie untersucht. Dadurch konnten wir zum besseren Verständnis und zur Definition von Leitlinien in diesem Zusammenhang beitragen. Dabei konzentrierten wir uns insbesondere auf die Untersuchung der Perzeption von Richtungsänderungen. Die Ergebnisse der Arbeiten wurden in [TK07] und [TSPK08] veröffentlicht.
2. Ein weiteres Gebiet ist die Analyse von Gruppen zweidimensionaler, zeitabhängiger Datenobjekte basierend auf einer Feature-Vector Repräsentation. Bei der Visualisierung zweidimensionaler, zeitabhängiger Daten werden häufig Trajektorien zur Abbildung der Datenveränderung über die Zeit eingesetzt. Diese Technik hat den Nachteil, dass die Zeichenfläche schnell überfrachtet wirkt. Um eine Stufe der visuellen Abstraktion zu schaffen, können die Objekte daher in Gruppen zusammengefasst werden und diese mit Hilfe verschiedener Visualisierungstechniken anstelle der einzelnen Objekte dargestellt werden. Um ein signifikantes Verhalten der Punkte und der Gruppen über die Zeit feststellen zu können, haben wir eine Reihe von wichtigen Objekteigenschaften definiert, um deren Entwicklung über die Zeit beobachten zu können. Dafür wurden von uns die zu untersuchenden Eigenschaften in drei Gruppen gegliedert: a) Beschreibung und Analyse einzelner Objekte und deren Bewegung. b) Untersuchung von Objektgruppen und Zusammenhänge zwischen Objekten und deren Gruppen. c) Die Analyse des Verhaltens verschiedener Gruppen untereinander. Die Auswertung dieser Eigenschaften erlaubt es, interessante Datenänderungen zu identifizieren und detailliert visuell zu explorieren. Die Ergebnisse wurden in [vLBR09] veröffentlicht.

-
3. Des Weiteren haben wir eine interaktive Technik zum visuellen Clustering von Trajektorien (zweidimensionaler, zeitabhängiger Daten) vorgestellt. Durch eine Gruppierung ähnlicher Trajektorien und die Auswahl eines jeweils repräsentativen Prototyps kann ein Überblick über den Datenraum geboten werden. Wir verwendeten zum Clustern der Daten ein neuronales Netz in Form einer Selbst-Organisierenden Karte (SOM). Wir haben dieses Verfahren, das auch zum Layout der Ergebnisse genutzt wird, in eine interaktive Visualisierung integriert. Dadurch kann das Training während des Analyseprozesses nicht nur beobachtet, sondern auch direkt beeinflusst werden und es kann ein besseres Verständnis des Vorgangs erreicht werden. Die Ergebnisse wurden in [STFK07], [SBTK08] und [SBvLK09] veröffentlicht.

Als Grundlage für die Entwicklung der neuen visuellen Analyseansätze, dienen die **Methoden von Visual Analytics und der damit verbundenen Forschungsbereiche**. Wir erweitern dabei die Klassifikation der Informationsvisualisierungs- und Interaktionstechniken, die in Visual Analytics verwendet werden.

1. *Klassifikation von Informationsvisualisierungstechniken*. Als zentraler Baustein des Gebiets Visual Analytics ist die Auswahl geeigneter Visualisierungstechniken von wesentlicher Bedeutung. Die Wahl hängt vor allem von der Art der Eingabedaten und der analytischen Aufgabe ab. Für die Kategorisierung von Visualisierungstechniken wurden in der wissenschaftlichen Literatur mehrere Datentypdefinitionen vorgeschlagen. Allerdings fällt die Einordnung komplexer Datentypen schwer, insbesondere für die in dieser Dissertation bearbeiteten Daten. Wir führen daher eine veränderte Klassifikation von Visualisierungstechniken auf der Grundlage einer neuen Definition des Datentypraumes ein. Neben wichtigen Visualisierungstechniken für die in dieser Arbeit untersuchten Graph- und 2D-zeitabhängigen Daten konnte so auch ein Überblick über sonstige, relevante Visualisierungstechniken gegeben werden. Diese Ergebnisse wurden teilweise in [vLKS*10] veröffentlicht.
2. *Klassifikation von Interaktionstechniken*. Derzeit werden in Visual Analytics jeweils unterschiedliche Taxonomien für Interaktionstechniken, die aus den Gebieten der Informationsvisualisierung, der Datenverarbeitung und der analytischen Beweisführung stammen, angewendet. Da Visual Analytics diese drei Bereiche integriert, ist ein bereichsübergreifender Ansatz für die Klassifikation der Interaktionstechniken erforderlich. Deshalb wurde eine neue, integrierte Taxonomie für die Interaktionstechniken im Rahmen der Visual Analytics durch die Vereinheitlichung und Erweiterung der Taxonomien der drei Bereiche erarbeitet. Diese Taxonomie ist in unserem State-of-the-Art-Report über die visuelle Graphenanalyse [vLKS*10] angewendet worden.

Die entwickelten Techniken können in verschiedenen **Anwendungsbereichen** wie z.B. Finanzen und Wirtschaft, Geographie, Sozialwissenschaften, Biologie, Transport oder Meteorologie angewandt werden. Wir demonstrieren unsere neuen Verfahren an zwei praxisrelevanten Datensätzen aus dem Bereich der Finanzanalyse. Zum Einen, handelt es sich um die Analyse von Beteiligungsstrukturen und zum Anderen, um die Auswertung von zeitabhängigen Risiko-Rendite-Daten.

Beteiligungsstrukturen beschreiben Verflechtungen zwischen Unternehmen in einer Volkswirtschaft. Sie können als gewichtete, gerichtete Graphen dargestellt werden, wobei Knoten die Unternehmen repräsentieren und Kanten deren Verknüpfungen darstellen. Die Analyse der absoluten Anteilseigner, der Controlling-Ketten und der Arten von Beziehungen zwischen zwei Unternehmen wird in diesem Szenario durch eine Kombination von anwendungsspezifischen, graphalgorithmischen Techniken in Kombination mit interaktiver Graphenvisualisierung unterstützt [TK08]. Eine Motivanalyse ermöglicht das schnelle Identifizieren interessanter Beteiligungsmuster, um z.B. spezielle Kontrollmuster zu erkennen. In einer Szenario-Analyse können mit Hilfe unseres Systems hypothetische Ereignisse wie Beteiligungsänderungen oder Firmenkonkurse simuliert werden. Mittels einer Motivanalyse können dabei potentielle Veränderungen in Kontrollstrukturen, die für die Unternehmensleistung wichtig sein können, identifiziert werden [vLGRS09]. Um die gesamte Wirtschaft zu untersuchen, können häufig auftretenden Unternehmenstrukturen durch Clustering-Verfahren identifiziert werden [vLGS09].

Zeitabhängige Risiko-Rendite-Daten werden in der Anlageanalyse an der Börse benutzt. Neben der Analyse einzelner Vermögenswerte ist auch die Auswertung der zeitabhängigen Änderungen von Gruppen von Vermögenswerten (z. B. in Länderentwicklungen) von Interesse. Wie unsere qualitative Benutzerstudie aufgezeigt hat, ist die kombinierte Darstellung von Querschnitts- und Zeitdimension sehr wichtig und führt zu neuen Einsichten in die Eigenschaften der Daten [TK07]. In unserer Arbeit unterstützen wir die Aktienmarktanalyse durch das Aufzeigen von außergewöhnlichen Bewegungen einzelner Aktien oder Gruppen von Aktien (beispielsweise Länderentwicklungen) [TK07, STFK07, SBvLK09, vLBRS09]. Diese Analysewerkzeuge können für eine breite Palette von Aufgaben im Finanzassetmanagement, etwa für die Erstellung von Anlageempfehlungen, für die Beurteilung der aktuellen Marktentwicklung oder für die Auswertung der historischen Trends verwendet werden.

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Visual Analytics	2
1.3. Visual Analysis of Weighted Directed Graphs	3
1.4. Visual Analysis of Two-Dimensional Time-Dependent Data	6
1.5. Thesis Structure	9
2. Visual Analytics and Related Research Fields	11
2.1. Introduction	11
2.1.1. Chapter Overview	11
2.2. Visual Analytics	12
2.2.1. Definition of Visual Analytics and its Relation to Other Research Fields	12
2.2.2. Interdisciplinary Aspects of Visual Analytics	14
2.2.3. Visual Analytics Process	16
2.2.4. Visual Analytics Research Trends and Challenges	19
2.3. Information Visualization	21
2.3.1. Taxonomy of Information Visualization Techniques	22
2.3.2. A New Definition of Data Type Space	23
2.3.3. Overview of Information Visualization Techniques According to the New Definition of Data Type Space	26
2.4. Interaction	33
2.4.1. Relevant Taxonomies of Interaction in Information Visualization, Reasoning and Data Processing	33
2.4.2. A New Unified Taxonomy of Interaction in Visual Analytics	36
2.5. Data Processing	39
2.5.1. Data Transformations	39
2.5.2. Data Mining	40
2.5.3. Application-dependent Data Processing Techniques	43
2.5.4. Use of Data Processing Techniques in Information Visualization	43
3. Visual Analysis of Weighted Directed Graphs	45
3.1. Introduction	45
3.1.1. Tasks	45
3.1.2. Contribution	47
3.1.3. Chapter overview	47
3.2. Background	48
3.2.1. Definitions	48
3.2.2. Algorithmic Graph Analysis	50
3.2.3. Graph Visualization	52

3.2.4.	Visual Graph Analysis	54
3.2.5.	Summary	56
3.3.	New Approaches to Visual Analysis of Weighted Directed Graphs	57
3.3.1.	Approach to Interactive Visual Exploration of Weighted Directed Graphs	57
3.3.2.	Approach to Visual Analysis of Graphs Using Motifs	58
3.3.3.	Approach to Visual Analysis of Many Graphs Using SOM Clustering	63
3.4.	Interactive Visual Exploration of Weighted Directed Graphs	67
3.4.1.	Introduction	67
3.4.2.	Interactive Visualization	67
3.4.3.	Visual Analytical Functions	68
3.5.	Visual Analysis of Graph Motifs	71
3.5.1.	Introduction	71
3.5.2.	Graph Motifs	71
3.5.3.	Interactive Definition and Visualization of Motifs	77
3.5.4.	Visual Analysis of Graph Changes using Motifs	79
3.5.5.	Visual Analysis of Graphs Using Motif-based Graph Aggregation	80
3.6.	Visual Analysis of Many Graphs Using SOM Clustering	87
3.6.1.	Introduction	87
3.6.2.	Graph Features for Measuring Graph Similarity	87
3.6.3.	Interactive Feature Selection and Visualization of Feature Space	88
3.6.4.	Interactive SOM Parameter Setting	89
3.6.5.	Interactive Visualization of SOM Clustering Results	89
3.6.6.	Interactive Visualization of SOM Clustering Quality	94
3.7.	Application	98
3.7.1.	Introduction	98
3.7.2.	Data	99
3.7.3.	Visual Exploration of Shareholder Networks	102
3.7.4.	Visual Analysis of Shareholder Networks based on Motifs	109
3.7.5.	Visual Analysis of Shareholder Networks using SOM Clustering	130
4.	Visual Analysis of Two-Dimensional Time-Dependent Data	137
4.1.	Introduction	137
4.1.1.	Tasks	138
4.1.2.	Contribution	138
4.1.3.	Chapter Overview	139
4.2.	Background	140
4.2.1.	Definitions	140
4.2.2.	Algorithmic Analysis of Two-Dimensional Time Series	141
4.2.3.	Visualization of Two-Dimensional Time Series	143
4.2.4.	Visual Analysis of Two-Dimensional Time Series	144
4.2.5.	Summary	145
4.3.	New Approaches to Visual Analysis of Two-Dimensional Time-Dependent Data	147
4.3.1.	Approach to Interactive Visual Exploration of Two-Dimensional Time Dependent Data	148
4.3.2.	Approach to Visual Analysis of Two-Dimensional Time Dependent with Grouping of Entities	149
4.3.3.	Approach to Visual Analysis of Two-Dimensional Time Dependent Data Using SOM Clustering	151

4.4.	Interactive Visual Exploration of Two-Dimensional Time-Dependent Data	154
4.4.1.	Introduction	154
4.4.2.	Interactive Visualization	154
4.4.3.	Perception Study for Visualization of Two-Dimensional Time Dependent Data Using Animation	157
4.5.	Visual Analysis of Two-Dimensional Time-Dependent Data with Grouping of Entities	163
4.5.1.	Introduction	163
4.5.2.	Time-Varying Features for Description of Groups of Two-Dimensional Time-Dependent Data Entities	163
4.5.3.	Interactive Visualization of Two-Dimensional Time-Dependent with Grouping of Data Entities	166
4.5.4.	Visual Analysis of Two-Dimensional Time-Dependent Data with Grouping of Entities	168
4.6.	Visual Analysis of Two-Dimensional Time-Dependent Data Using SOM Clustering	171
4.6.1.	Introduction	171
4.6.2.	Similarity Measures and Transformation for Two-Dimensional Time-Dependent Data	171
4.6.3.	Interactive Feature Selection and Visualization of Feature Space	174
4.6.4.	Interactive Visualization and Control of SOM Clustering Process	176
4.6.5.	Interactive Visualization of SOM Clustering Results	178
4.6.6.	Interactive Visualization of SOM Clustering Quality	187
4.7.	Application	192
4.7.1.	Introduction	192
4.7.2.	Data	193
4.7.3.	Visual Exploration of Time-Dependent Risk-Return Data	194
4.7.4.	Visual Analysis of Time-Dependent Risk-Return Data With Asset Grouping	197
4.7.5.	Visual Analysis of Time-Dependent Risk-Return Data using SOM Clustering	210
5.	Conclusions and Future Challenges	217
5.1.	General Remarks	217
5.1.1.	Conclusions	217
5.1.2.	Future Challenges	218
5.2.	Visual Analysis of Weighted Directed Graphs	219
5.2.1.	Conclusions	220
5.2.2.	Future Challenges	220
5.3.	Visual Analysis of Two-Dimensional Time-Dependent Data	221
5.3.1.	Conclusions	221
5.3.2.	Future work	222
A.	Publications and Talks	225
A.1.	Publications	225
A.2.	Talks	226
B.	Curriculum Vitae	227
	Bibliography	229

1. Introduction

1.1. Motivation

The assessment of large amounts of data is important in many application areas including finance and economics, biology, transportation, and the social sciences. In particular, in the international financial markets, data providers such as Bloomberg and Thompson Reuters offer access to large pools of data taken in real time from a variety of sources, for instance, from stock exchanges' electronic trading systems or from national statistical institutes. Financial analysts need to effectively assess these large amounts of data in order to make good investment decisions or to offer relevant financial consulting services.

The general progress made in the field of information technology has altered the functioning of financial markets. Two important aspects are changing: market integration is improving and price formation is becoming more information dependent. These aspects imply that the analyst need to leverage as much information as possible in order to face the significant competition. Thus, analysts need to arrive at the best possible decisions in a short span of time in order to benefit from business opportunities.

The key issues concern analysis of financial data observations. They are, in general, identified on the basis of at least two main characteristics:

- *the entity* or the financial instrument to which the observation refers, and
- *the point in time* of the observation.

For instance, these two main characteristics are reflected in the statement “On 10 July 2009 the Siemens share noted at USD 65.22”.

The first issue centers on the *relationship between entities*. Indeed, in a modern economy large structures of cross-company shareholding relationships exist forming complex structures of investment and control between the involved entities within a corporation. These corporations, themselves, are in turn entities within the wider network of the economy. The analysis of large corporate shareholding networks is an important task in the domains of corporate governance, financing, and financial investment. For example, according to the European Corporate Governance Institute report [Eur07], depending on the type of structure, between 58% and 92% of investors take the presence of control enhancing mechanisms such as pyramids and cross-ownerships into account in their investment decisions.

The second issue focuses on *time dependence*. The analysis of the evolution of financial indicators of equities through time is central to financial investment decisions. In this respect, the analysis of correlations between risk and return is amongst the most popular ones. The return measures the yield of the stock (dividend plus the change in the price of the stock relative to the price in the previous period). The riskiness of future returns is proxied by the volatility of the return (the variability of the expected return). It means that the higher the variability, the higher the risk associated with the stock. Stocks with higher returns having the same level of volatility are preferred. As risk and return indicators of assets usually vary over time, the assessment of the 2-dimensional observations along the time axis is considered important in addition to examination of indicators along the cross-section of the data at any given point in time. These analytical results may lead to the need for portfolio adjustments in order to enhance future asset returns.

These two analysis strands examine shareholder relationships and time-varying risk-return data, which are examples of two common data types: *weighted directed graphs* and *two-dimensional time dependent data*. Therefore, tools for analyzing and processing of this type of information need to be developed.

Analytical tasks for the data are rarely completely well-defined in advance. Rather, hypothesis generation and verification drives the analysis process. Therefore, *user involvement in the analytical process* plays a central role. The analyst must be provided with flexible integrated tools, which she may use on demand for accomplishing her task. The control of the iterative analytic process in each step and the possibility to engage in feedback loops are necessary. This makes further development of the systems offering *visual analysis features* combined with *effective data processing* under user control is necessary in order to cope with the higher demands of the analysts.

1.2. Visual Analytics

Modern data processing and information visualization systems are key elements in visual analysis of weighted directed graphs and two-dimensional time-dependent data. “However, a simple combination of visualisation with computational analysis is not sufficient. The challenge is to build analytical tools and environments where the power of computational methods is synergistically combined with human’s background knowledge, flexible thinking, imagination, and capacity for insight.” [AAK*08] This is the goal of Visual Analytics, a new interdisciplinary field which makes use of techniques from data processing, data visualization, human-computer interaction, reasoning and other fields to help gaining insight into data (see Figure 1.1).

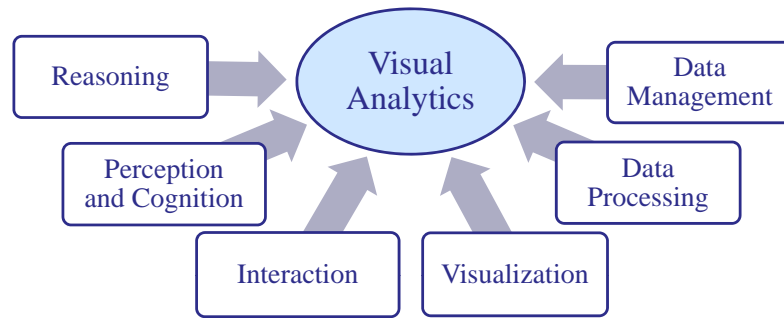


Figure 1.1.: Scope of Visual Analytics.

Visual Analytics research is still a young discipline. Therefore, we focus on its **methodical aspects** (see Chapter 2). We discuss and extend the relevant methodologies in the related fields, while taking the Visual Analytics perspective. **Contributions** are made with regard to the following aspects.

1. *Classification of information visualization techniques.* As a central building block of Visual Analytics, the selection of appropriate visualization techniques is essential. The choice is mainly based on the type of input data and the analytical task at hand. For the classification of visualization techniques, several data type definitions are proposed in the academic literature. However, these definitions can not unambiguously be applied to complex data types, in particular to the type of data used in the thesis. We therefore introduce an altered classification of visualization techniques based on a new definition of the data type space (see Section 2.3.2). In this way, we can also provide an extensive survey of the main information visualization

techniques including those relevant for the visual analysis of the two data types in the focus of the thesis (see Section 2.3.3). This work has been partly published in [vLKS*10].

2. *Classification of interaction techniques.* At present, separate taxonomies for interaction techniques applied in Visual Analytics exist in information visualization, data processing and reasoning. However, as Visual Analytics integrates these three areas, a common approach to the classification of the interaction techniques is needed. Therefore, we elaborate a new, integrated taxonomy for the interaction techniques in the context of Visual Analytics by unifying and extending the taxonomies of the three areas (see Section 2.4.2). This taxonomy has been used in our state-of-the-art-report on visual graph analysis [vLKS*10].

1.3. Visual Analysis of Weighted Directed Graphs

The analysis of graphs is an important element in many application areas such as transportation, sociology, biology, finance, or software engineering. For example, in corporate governance, financing and financial investment, the analysis of shareholding networks is an important task.

Graphs describe relationships between entities. They consist of vertices (nodes or entities) and edges (relationships between entities). A **graph** is a pair $G = (V, E); E \subseteq [V^2]; V \cap E = \emptyset$, where elements of V are vertices and elements of E edges [Die05]. The main categorization of graphs is according to the direction of the edges into directed and undirected. Additionally, graphs are distinguishable according to their edge attributes into weighted and unweighted or according to the node properties into labeled and unlabeled (see Section 3.2.1 for more elaborate discussion of graph types). Graphs can include cycles (i.e., closed paths – ordered sets of vertices following the edge directions). We regard weighted directed graphs having only a maximum of one edge between the ordered pair of vertices (i.e., networks). If not stated otherwise, we consider static graphs without time-dependence. The graphs considered in the thesis can include cycles.

The accomplishment of any graph **analytical task** faces as a prerequisite the need to properly understand the global and local graph structures, entity relationships and the structure of the interlinkages between entities. This understanding can either concentrate on the examination of the entity relationships in one graph, on the identification of frequent or interesting substructures (motifs) occurring within a graph, or on the assessment of similarities and of differences among many graphs (see Figure 1.2). These tasks are in the focus of this thesis. A more detailed discussion of these graph analytic tasks is presented in Section 3.1.1.

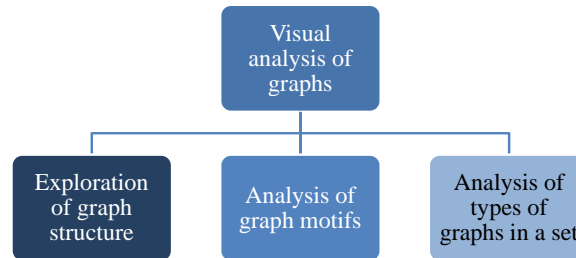


Figure 1.2.: Selected visual graph analysis tasks in the focus of the thesis.

In the visual analysis of graphs, the research focus has been set on the development of efficient presentation and navigation techniques. However, in order to support the analytical tasks, an enhancement of these techniques with graph analytical algorithms in an interactive way is deemed beneficial.

We present novel techniques for supporting visual analysis of graphs while concentrating on the above-mentioned tasks. The main **contributions** are (see also Chapter 3):

1. We support the exploration of graphs, focusing on tasks in the analysis of connections and paths, by efficiently combining *algorithmic graph analysis and interactive graph visualization*. We specifically adapt the approach for the analysis of shareholder networks. The results were published in [TK08].
2. We combine *graph visualization with the detection of graph substructure* (i.e., motifs). In relation to state-of-the-art approaches, our approach includes the visualization of both predefined and user-defined motifs found in the graph (see Figure 1.3).¹ We enhance the analysis of graph changes on the entire graph structure. We propose a hierarchic graph aggregation based on motifs. Graph aggregation brings relationships between the local graph substructures to the fore. A flexible combination of these techniques provides tools for the detection of (novel) graph substructures at various levels of abstraction. The results were published in [vLGRS09].
3. We introduce *clustering of a large number of graphs*. Specifically, we consider a set of weakly connected components of a graph. Although many graph visualization and aggregation techniques exist, visual feature-based clustering of connected components for analysis of graph similarities has not been examined before. The clustering provides an overview of the typical graph structures occurring in the data (see Figure 1.4). Interactive visual interfaces provide for various views on the raw data, for the selection of clustering parameters and for the exploration of the results including their quality. The results were presented in [vLGS09].

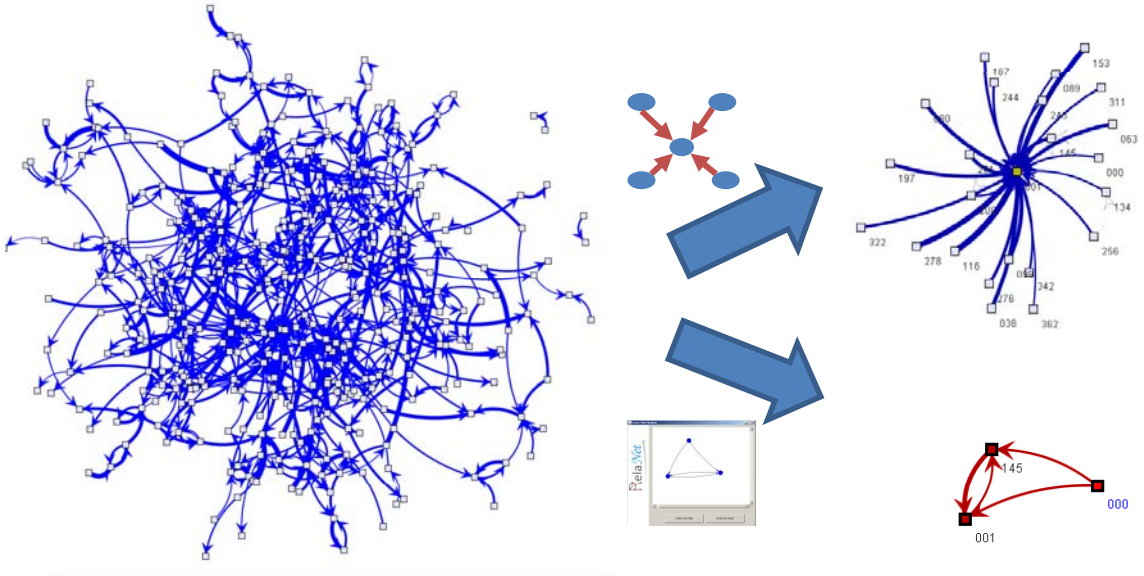


Figure 1.3.: An example for the use of visual exploration of graph motifs. Left: Node-link visualization of the whole graph. It shows an overcrowded display. Right: The result of the new approach showing selected motifs found and filtered in the original graph. This view reveals interesting substructures of the graph in a more interpretable way.

¹Graph motif search is a NP-hard problem.

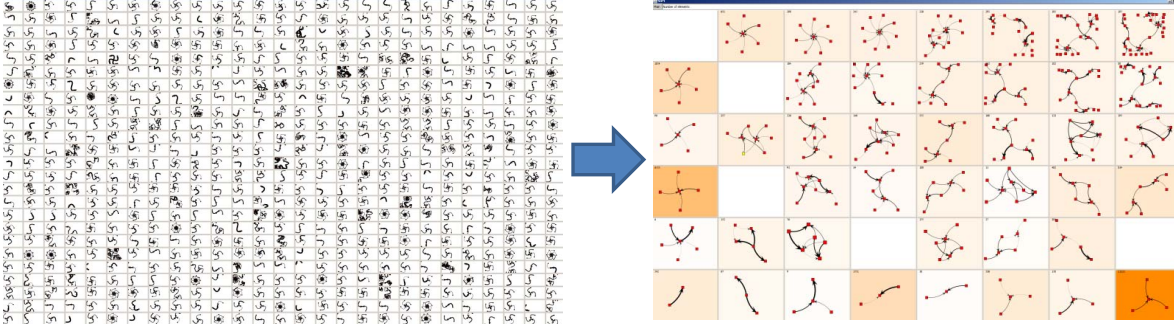


Figure 1.4.: Illustration of the clustering of multiple graph components revealing types of graphs in a set. Left: A small part of the set of graph components (ca 1% of 40,000 components) using state-of-the-art graph visualization techniques. This view does not support assessment of types of graphs and their similarities. Right: Visualization of the result of clustering of multiple graph components using our approach. It shows types of graphs and their frequency (represented by background color) in the analyzed data set.

We **apply these techniques** on real-world corporate shareholding structures data (see Section 3.7). Shareholding relationships between companies in an economic system can be regarded as a weighted directed graph with nodes representing companies, and weighted, directed edges representing the “holds-shares-in” relationship between firms. In this respect, we support the analysis of ultimate shareholders, controlling chains and relationships between two companies by a combination of application-specialized graph algorithmic techniques with interactive graph visualization [TK08]. Additionally, the motif-based analysis allows to detect interesting shareholding patterns, such as control enhancing mechanisms. The what-if-analysis of buying/selling shares or of company defaults is enhanced by the detection of new/changed controlling structures relevant for the corporate performance [vLGRS09]. When analyzing shareholding relationships in the whole economy, clustering methods help to reveal the types of company structures often prevailing in the whole economy [vLGS09].

1.4. Visual Analysis of Two-Dimensional Time-Dependent Data

The study of developments of two-dimensional time-dependent data (while considering the relationship between the two dimensions) is relevant in a variety of domains. For example, technical, geographic, meteorological, biologic as well as financial and economic applications deal with these data types. The application data set may result from direct measurement (e.g., risk-return, temperature-humidity, geographic position) or of projection methods from higher-dimensional data spaces into two dimensions.² Additionally, in geographic data sets, the locations can be represented in this manner. In the two-dimensional data sets, measurements are often time-dependent leading to the derivation of so-called time series of observations.

Two-dimensional time-dependent data are an extension of static two-dimensional data points with the time dimension. Time is an implicit additional dimension of the two-dimensional data items. One two-dimensional time series of an entity k can be **defined** as: $T^k = \{t_0^k, t_1^k, \dots, t_n^k\}$, an ordered set of points, where t_i^k is the 2D position $t_i^k = [x_i^k, y_i^k]$ of the entity k at time point i , where $i \in I, I = \{0, \dots, n\}$. Note that further related definitions can be found in Section 4.2.1. In our work, we assume equidistant time steps, i.e., the data points have been measured at regular time intervals and linear movement of the points between the time steps with constant speed. If appropriate for the analysis, the points have additional information on their class labels and weight. When analyzing groups of entities, we also regard class labels of the data items. We assume that the class labels stay constant over the analyzed time span.

Typical **analysis tasks** on static two-dimensional data include assessment of the overall structure and distribution of the data, assessing the spatial relationship between data elements, and the identification of clusters and outliers. By extending static data with a time dimension, the examination of *dynamics* of individual and grouped points³ is also brought into the focus of the analysis. In our work, we concentrate on the examination of the dynamics of individual data points, on the analysis of dynamics of groups of entities (and entities within groups), and on the analysis of movement patterns of the entities (see Figure 1.5). Section 4.1.1 discusses these tasks in more detail.

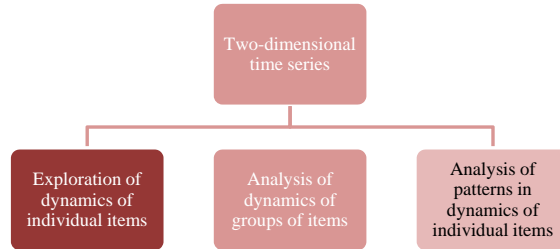


Figure 1.5.: Selected tasks in the analysis of two-dimensional time-dependent data in the focus of the thesis.

With regard to two-dimensional time series, there is a need to study possible interactive visualization techniques and their underlying perception guidelines. For the analysis of the dynamics of groups of entities (and entities within groups), the current methods for visual analysis of distributions of static data need to be extended with a focus on time-dependence. Moreover, the visual analysis of movement patterns in two-dimensional time-dependent data has extensively been studied mainly in the context of geographical data. The analysis of data from other domains opens new challenges. Although the techniques to undertake the analysis of two-dimensional time-

²We regard the two dimensional data as given disregarding the possible loss of data quality through the external projection step.

³The time development of each group of entities creates a complex composition of movements of individual group members.

dependent data exist either with regard to the visualization or to data processing function, they have as yet not been integrated in a satisfactory manner.

In our work, we make the following **contributions** to the visual analysis of two-dimensional time-dependent data (see also Chapter 4).

1. We support the visual exploration of two-dimensional time-dependent data by interactive animation and trajectory visualization. The effectivity of animated visualization depends on the animation settings which need to follow the human perception capabilities. We therefore conduct a *study on the perception of animation* in the visualization of time-dependent 2D points. In this way, we contribute to the better understanding of perception effects and provide guidelines for the use of animation in scatterplot-based information visualization. In relation to previous studies, we concentrate on the perception of direction changes for groups of points. The results of the individual research parts were published in [TK07] and [TSPK08].
2. We extend the *interactive visualization of time-dependent groups of entities with feature-based data analysis* in the time-dependent domain. The previous studies mainly concentrate on visualization of group data or visual analysis of static data distributions. The data visualization easily gets overcrowded and analysis of static data needs to consider also the time dimension of the data. We propose the identification and the presentation of interesting data developments by monitoring a set of relevant data features for groups of points over time (see Figure 1.6). In this respect, we also extend the set of previously proposed static features with new features suitable for analysis of time-dependent groups of entities. The results were published in [vLBR09].
3. We introduce *visual clustering of trajectories* (2D time-dependent data) using self-organizing maps (SOM). SOM-based trajectory clustering provides an overview of the typical data movements and allows to identify extraordinary data developments. In this respect, we closely combine interactive visualization with data processing methods (see Figure 1.7). Furthermore, we enhance this approach with user-driven visual monitoring of the clustering process and interactive initialization of the algorithm. This allows for better understanding of the clustering results and the inclusion of possible prior knowledge in the analysis. The results were published in [STFK07], [SBTK08] and [SBvLK09].

We **apply** these techniques to the analysis of developments of multiple assets on the stock market, i.e., time varying risk-return data (see Section 4.7). In addition to the analysis of developments of individual assets, also the assessment of time-dependent movements of groups of assets (e.g., country developments) is of interest. As our qualitative user study shows, the combination of cross asset and time dimension analysis is very important and leads to new insights into the data [TK07]. In our work, we enhance stock market analysis with a tool for detecting common and exceptional stock market movements of single assets and country-grouped assets [TK07, STFK07, SBvLK09, vLBR09]. These analytical tools can be extensively used for a wide range of tasks in financial asset management such as providing investment recommendations, assessment of current market movements, or examination of historical market trends.

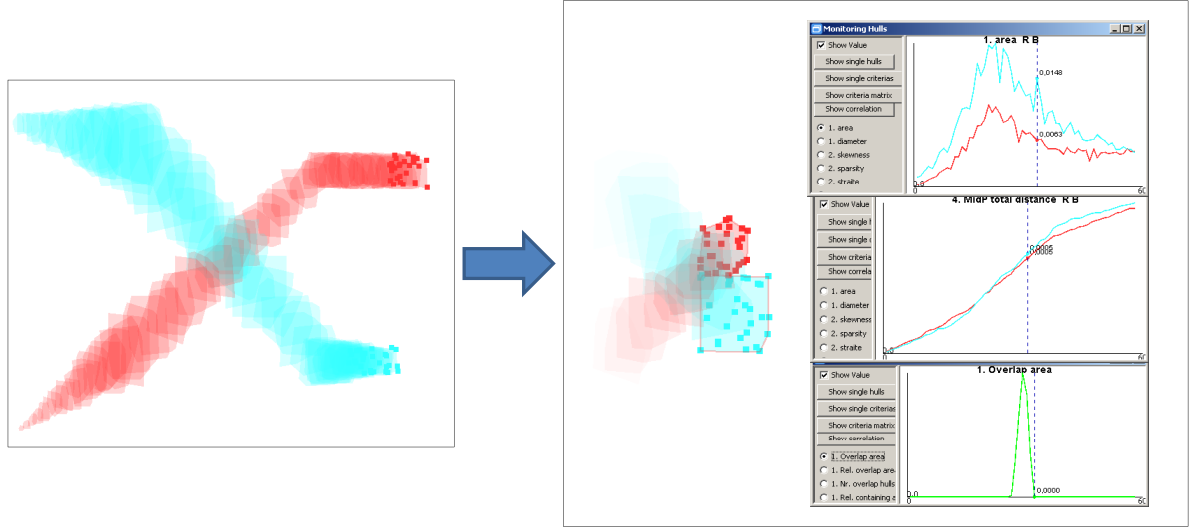


Figure 1.6.: Illustration of visual analysis of groups of two-dimensional time-dependent entities. Left: State-of-the-art presentation of the dynamic grouped data using convex hull trace visualization. It shows an overcrowded display that is difficult to explore. Right: The new feature-based visual analysis revealing interesting views on the data. The main view (left part) shows the detected interesting data behavior. Additional views show the time-dependent values of the extracted features indicating interesting time periods.

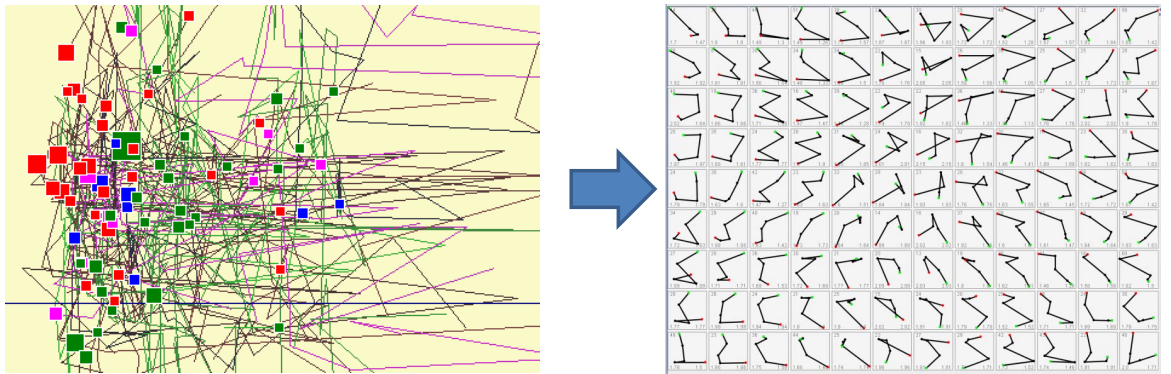


Figure 1.7.: Illustration of the new visual clustering of trajectories using SOM. Left: State-of-the-art trajectory visualization of the data. It shows an overcrowded display that is difficult to analyze. Right: The result of the new approach using SOM-based clustering presents an overview of the trajectory patterns occurring in the data set.

1.5. Thesis Structure

The thesis is structured as follows:

Chapter 2 introduces the *broad theoretic* background for the work presented. It defines and describes the field of Visual Analytics against the background of related research areas. Visual analytics is presented as an interdisciplinary field combining mainly information visualization, interaction and data processing techniques. Subsequently, the methods used in these three areas are systematized and new perspectives on these areas are presented. The detailed description of studies in these areas focusing on particular problems relevant to the thesis is then presented in the subsequent two chapters together with the specific problems addressed.

Chapters 3 and 4 describe the *new methods for visual analysis of weighted directed graphs and two-dimensional time series*, respectively. Both chapters follow the same structure. The introduction motivates, describes the tasks tackled and the respective scientific contribution of each chapter. The background section reviews relevant studies in the areas of visualization, data processing and Visual Analytics. The following conceptual section offers a broad overview of our approach before dealing, in more detail, with individual aspects of the methods used. Our approach starts, as a basis, from an (enhanced) interactive information visualization perspective and then continues with two methods combining interactive visualization with two different types of data processing – on the one hand, feature/motif based techniques and on the other hand clustering techniques. Each sub-approach thereby tackles one task area for the respective data type. Figure 1.8 shows an overview of the thesis structure indicating the respective subsection of the chapter. Owing to the complexity of the issues and their technical specificities, the detailed descriptions of the implementation of the three approaches are then presented in following three sections. The application section highlights the effectivity of the three methods for a real-world data set (shareholding networks and risk-return data, respectively).

Chapter	Approach Data type	Enhanced interactive information visualization	Specific processing-based visual analysis	Clustering-based visual analysis
3	Weighted directed graphs	Visualization methods enhanced with relationship analysis	Motif-based visual analysis	Clustering-based visual analysis
4	Two-dimensional time dependent data	Visualization methods and perception study	Feature-based visual analysis	Clustering-based visual analysis

Figure 1.8.: Overview of the structure of the Chapters 3 and 4 which discuss visual analysis of two distinct data types (see also Figures 1.2 and 1.5).

Chapter 5 stresses the *key findings* from the conceptual work and its subsequent implementation on real world data and outlines possibilities for *future work*. It firstly discusses them at a general level and then focuses more specifically on the two main parts: visual analysis of weighted directed graphs and of two-dimensional time-dependent data.

2. Visual Analytics and Related Research Fields

2.1. Introduction

In this chapter, we present a broad overview and a systematization of the techniques used in Visual Analytics of abstract data. We first compare Visual Analytics, with adjacent research fields (Visual Data Mining and Knowledge Discovery in Databases), as these areas are often not clearly defined owing to their dynamic evolution. Visual Analytics is seen as an interdisciplinary field. Therefore, subsequently, we discuss three separate areas (information visualization, interaction and data processing) relevant to Visual Analytics. For the classification of the available information visualization techniques applicable also for complex data types, we present a new definition of data type space. Using this classification scheme, we provide an overview of the main information visualization techniques used in Visual Analytics tools. For the visual analytic purposes, we unify taxonomies of interaction techniques from information visualization, data processing and reasoning. We use the techniques and methods introduced here as a basis of conceptualization of approaches introduced in the following two chapters.

2.1.1. Chapter Overview

In this chapter, we first introduce Visual Analytics and compare it to related research areas (see Section 2.2). Visual Analytics is an interdisciplinary field, therefore we describe this aspect before dealing with models of the Visual Analytics process. We finalize discussing Visual Analytics with a review of main current Visual Analytics research trends and challenges.

Next, we then provide a description of information visualization (see Section 2.3). We present a review of a variety of taxonomies and introduce a new taxonomy of information visualization techniques based on data type. We use this new data type space definition for classification of information visualization techniques. This classification provides an overview of available visualization techniques for usage in Visual Analytics systems.

The third section (Section 2.4) provides an overview of interaction techniques and the respective methodologies. After introducing the methods in three fields relevant to Visual Analytics (information visualization, data processing and reasoning) we propose a unified model of interaction techniques specifically for Visual Analytics.

Finally, the section on data processing methods (Section 2.5) summarizes methods for computer algorithmic analysis of data suitable for Visual Analytics purposes. It discusses data transformation and data mining techniques in particular. After introduction of types of data mining methods, we concentrate on clustering (self organizing maps in particular), as it encompasses a broad variety of methods for unsupervised examination of large amounts of data by revealing structures and observing characteristics of groups of data offering output for visual exploration used in this thesis.

The work presented in this chapter is partially based on the following publications [TK07], [TS08], [vL-BRS09], [KTSZ08], [STFK07], [TKSK08] and [vLKS*10].

2.2. Visual Analytics

2.2.1. Definition of Visual Analytics and its Relation to Other Research Fields

Visual analytics research field evolved from Information Visualization and Scientific Visualization [KMS*08]. It has effectively started to grow after the publication of the seminal book by Thomas and Cook in 2005 [TC05]. Therein, **Visual Analytics** (VA) is defined as follows.

*Visual analytics is the science of analytical reasoning facilitated by
interactive visual interfaces.
Thomas and Cook [TC05]*

The above definition embodies several key elements: *analytical reasoning* and *interactive visual interfaces* that are also to be found, for instance, in the definition of **visual data mining**. In fact, this is characterized as “*the process of interaction and analytical reasoning with one or more visual representations of abstract data that leads to the visual discovery of robust patterns in these data that form the information and knowledge utilized in informed decision making*” [SBM08] .

This definition seems to differ from the first characterization presented above, with respect to its emphasis on the discovery of patterns in “abstract data”. However, when looking at a second, more explicit, definition of Visual Analytics proposed by Keim et al. [KAF*08, KMS*08] a significant overlap between the two fields of research becomes apparent.

*Visual analytics combines automated analysis techniques with interactive visualizations for an
effective understanding, reasoning and decision making on the basis of very large and complex
data sets.
Keim et al. [KMS*08]*

At the same time, this proximity is not limited to these two fields – indeed Ankerst [Ank01] defines Visual Data Mining (VDM) as “*a step in the Knowledge Discovery in Databases (KDD) process that utilizes visualization as a communication channel between the computer and the user to produce novel and interpretable patterns.*” - thus linking VDM and, by implication, VA to the KDD process.

Comparison of the relevant fields of research (VA, VDM and KDD) On the basis of the above definitions and the associated literature, we summarize a number of key aspects with respect to the relevant fields of research (VA, VDM and KDD) in the Table 2.1. This allows gaining an overview of the salient features of the fields such as goals, data sets applied, tools used and user involvement, in order to conduct a comparison.

A number of observations follow with respect to the rows of the table:

1. The *goals* encompass gaining knowledge from or insight into the data, confirm hypothesis and discover new meaningful patterns in the data.
2. The *input data* may be of various kinds and is often large/massive/vast/huge. A closer look at the literature however does not help to specify what large/massive/vast/huge means. The explanations are either not given or vary along relevant works (see also [FdOL03]).
3. Strong *user involvement* in the iterative process seems to be in the focus of all disciplines. The user role may vary from determining analysis and visualization steps to full steering of the computation and visualization. In this respect, a similarity to computational steering [CW01] is apparent.

	Visual Analytics (VA)	Visual Data Mining (VDM)	Knowledge Discovery in Databases (KDD)
Goal	<p>[TC06] help the analyst discover unexpected and missing relationships that might lead to important insights, [TC05, KAF*08]: Synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data. Detect the expected and discover the unexpected.</p> <p>[Cha09] everyday data, [TC06] massive, complex, dynamic and often conflicting data, [KAF*08] very large and complex data sets, data overload</p>	<p>[SBM08]: to help a user to get a feeling for the data, to detect interesting knowledge, and to gain a deep visual understanding of the data</p> <p>[KW04] vast amounts of data</p>	<p>[FPsS96]: verification of user's hypothesis, discovery of new patterns (discovery can be prediction and description)</p> <p>[FPsS96]: large amounts of data, data overload</p>
User role	[TC06] tight involvement of the user in the process	[KW04] tight integration of human in the data mining process	[FPsS96] the process is interactive and iterative, involving numerous steps with many user decisions
Tools and interdisciplinarity	[TC06] combination of analytical reasoning, visual representations and interaction, data representation and transformations and production, presentation and dissemination techniques; [KMS*08] combination of KDD, statistics, mathematics and human perception, reasoning resp. combination of automatic analysis with human background knowledge and intuition; interaction, cognitive and perceptual science, presentation, production and dissemination, data management and knowledge representation, knowledge discovery [KAF*08] combination of automated analysis, interactive visualization and reasoning. It integrates Information and Scientific Visualization with Data Management and Data Analysis Technology, as well as Human Perception and Cognition research.	<p>[KW04]: combination of automatic data mining and interactive visualization; applying human perceptual abilities;</p> <p>[SBM08] combination of data mining, interactive visualization, analytical reasoning, data transformations.</p>	[FPsS96] intersection of data mining, machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, data visualization, and high-performance computing.

Table 2.1.: Summary of key aspects of Visual Analytics, Visual Data mining and Knowledge Discovery in Databases.

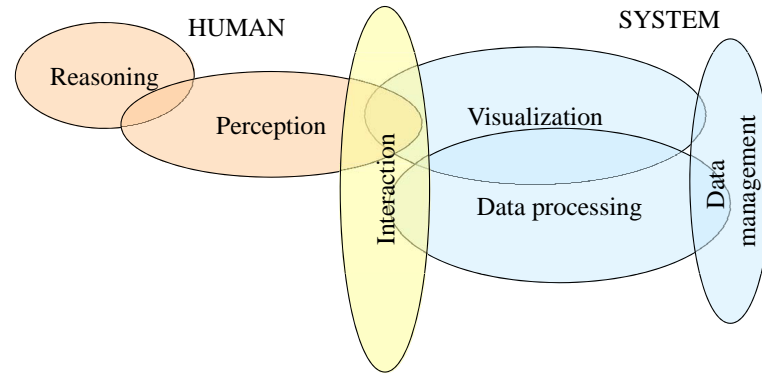


Figure 2.1.: Interdisciplinary fields of Visual Analytics and their relationship.

4. All fields use *multidisciplinary integrated tools*. The techniques in all fields include data visualization and data analysis. Moreover interactivity, knowledge discovery/management play a role. Other relevant techniques (such as database technologies, high performance computation, etc.) are used on demand. A closer look at the historic development of the fields reveals slight differences in the strengths of focus on the single disciplines. For example, Visual Analytics literature focuses more on techniques of interactive visualization and KDD on data mining/data analysis.

Overall the above table confirms the *similarity of all fields of research (VA, KDD and VDM)* considered by these features. The difficulty when comparing the fields is in *different wordings and possible different meanings of the same terminology* used in the articles and even within each article. Only sporadically, the authors define, what they mean when using the terminology (e.g., [BL09]).

2.2.2. Interdisciplinary Aspects of Visual Analytics

Visual Analytics (as well as VDM and KDD) makes use of techniques from various fields including data analysis, data visualization, human-computer interaction, perception and cognition, and others while finding synergies between human reasoning and perception and cognition skills and computer computation capabilities (see Figure 2.1 for an illustration)

On the computer side, VA relies on the combination of visualization with data processing building on efficient data management. There is a strong overlap of these fields as, e.g., on one hand visualization uses data processing and data management as a pre-processing tools and on the other hand data processing field uses visualization for presentation of the results. Interaction (in particular using visual interfaces) provides the means of communication between human and computer.

On the human side, the perception and reasoning are important factors for Visual Analytics research. Perception (and cognition) facilitates awareness of the sensory information on the interactive visual interface. Human reasoning helps to make sense out of the presented information. Only an appropriate combination of these factors leads to efficient gaining insights from the data. Therefore the synergetic study and research in these areas is necessary in Visual Analytics. In the following, we explain the role of the individual fields in more detail with respect to Visual Analytics.

Data management Data management helps to effectively store and extract suitable input data for the analysis. Recent data management concentrates not only on data storage and access but also on data integration and cleaning [KAF*08].

Data processing The roles of data processing (computer based data analysis) methods are to a) automatically detect interesting patterns in the data, verification of user hypothesis or discovery of new patterns [FdOL03, FPsS96] and b) perform effective transformations to convert data into new meaningful forms [TC06]. The degree of automation of the algorithms varies significantly from fully automatic to user-guided. In general the algorithm searches for patterns, not the human (which contrasts to sole visualization approaches). However, it is the human who is able to interpret and understand the found patterns.

Data processing (analysis) techniques include mainly data mining (e.g., clustering, regression), data transformation techniques (e.g., data reduction, dimension reduction, feature extraction), and application dependent techniques (e.g., economic analysis). Please note that in this thesis, the term “data mining” as used by Keim and Fayyad [FdOL03, KAF*08] and data transformation as used by Thomas and Cook [TC06] is generalized into the term “data processing”.

Visualization In general, the main value of visualization is gaining insight into and understanding of the underlying data [Spe07]. In contrast to data analysis methods, in using visualization, the human searches for possibly interesting patterns in the data, not the automatic computer processing tools.

According to Keim [KMS*08], there are three main roles of visualization in the analysis process: result presentation, confirmatory analysis and exploratory analysis (see also Figure 2.4 on page 18). The presentational function of visualization is to effectively display results of an analysis. In confirmatory analysis, interactive visualization is used for supporting user hypothesis about the data. In exploratory analysis (also called exploratory data analysis or EDA), interactive data display allows the user to search in the data space for potentially useful information and formulate hypothesis without dependence on a-priori assumptions. These insights may be used for formulating further (automatic) analysis steps. In this respect, visualization supports both modes of sense-making (see reasoning section below).

In addition, Ferreira proposes [FdOL03] that visualization may be used to help users to understand how data analysis algorithms works. This aspect is particularly important in connection to data mining.

Interaction Interaction in respect to Visual Analytics includes aspects from information visualization data processing and reasoning. It includes exploration and navigation of data space (information visualization), capturing user insights and tracking of analytic activity (reasoning) and interaction with data mining tools, etc. [War00, GZ08, YKSJ07, BL09]. The main role of interaction thereby is to intervene in the analytics process, e.g., by changing of view on the data leading to new insights or interpretations. According to Ware [War00], interactive visualization includes feedback loops of 3 classes from lowest to highest level:

1. *data manipulation loop*: objects are selected and moved using eye-hand coordination
2. *exploration and navigation loop*: finding way in the data space and thereby building a mental model of the data
3. *problem solving loop*: analyst forms hypothesis about the data and refines them through an augmented visualization process

Although this explanation was primarily meant with regard to information visualization, it applies also to Visual Analytics. The key aspect thereby is the distinction between the levels of interaction including the loop

which leads to new knowledge. This type of loop can be considered as the sense making interaction loop and thereby directly connected to reasoning in the sense of Visual Analytics (see reasoning below).

Perception Perception studies provide guidance for efficient data presentation and interaction in order to maximize gained insight from the data [KAF*08]. In this respect mainly information visualization part of Visual Analytics makes use of the results gained from studying human visual (and haptic) perception.

Reasoning Reasoning, in the sense of Visual Analytics, means applying human judgment to reach conclusions from a combination of evidence and assumptions [TC06]. The study of how humans gain new knowledge from the data is in the focus of this research area. The basis forms the so-called “sense-making loop” [PSC05] containing two main ways of analytic process: a) bottom up (from data to hypothesis and presentation) or b) top down (from presentation and hypothesis to data confirming the hypothesis). This process is not straightforward but includes many sub-loops. For studying how the process is performed and what results are gained, capturing of the analytic steps and annotation of the interpretations is used [GZ08]. In this respect, Visual Analytics partly overlaps with **knowledge management** methods in the role of capturing insight gained throughout the analysis, process and presenting externalized knowledge (knowledge visualization).

2.2.3. Visual Analytics Process

The currently mostly used model for Visual Analytics process (VA process) has been introduced by Keim et al. [KAF*08] (see Figure 2.2). On an abstract level it describes the way from input data, their pre-processing via interactive visualization and model building loops to the insight. The feedback loop provides the possibility of repeating the process, for example, for analysis of new data, new hypothesis etc.

An important difference to data mining, information visualization and KDD pipelines (see e.g., [CMS99], [HK06], [FPsS96]) is a) the direct inner-loop between visualization and data models and b) no strict process steps by possibility of both either direct data visualization or data mining. In all other models the data input is automatically processed (sometimes in various steps including data mining) and either subsequently displayed (in information visualization pipeline) or leads directly to knowledge (in KDD process).

The presented visual analytic process is very similar, although not seen at the first sight, to the visual data mining process presented by Simoff et al. [SBM08] (see Figure 2.3). Both processes tightly combine visualization and data mining in several loops under the user control. There are no specific linear process steps to be followed. Data pre-processing can be used in the initial stage, data mining and interactive visualization in further loops. Tight user control via steering of the loops is important. Keim however explicitly indicates by the two interaction arrows between visualization and models their tight connection and possibility of several inner loops (e.g., visualization of intermediate results), which is the major difference between the models.

Both models refer to *data mining* as an integral part of the process. In the VDM process, data mining can, but does not need to, be used to support the analytical process. Data mining can be applied either before or after the interactive visualization of the data. Although not seen at the first glance, this is very similar to the visual data mining model of Keim et al. [KW04], presented earlier, where the inner loop between visualization and data mining provides the same analytic possibilities (see Figure 2.4). Recently the tight integration of visualization and data mining for knowledge discovery has been discussed by Bertini et al. [BL09] showing the benefits from visualization and data mining in knowledge discovery (see Figure 2.5 top). Their model also includes feedback loops between data, model building and visualization (see Figure 2.5 bottom).

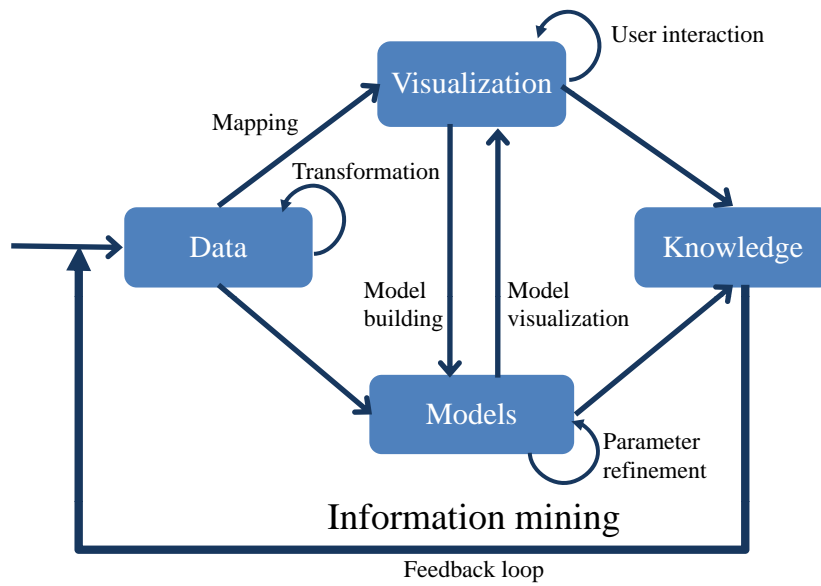


Figure 2.2.: Visual analytics process by Keim et al. [KAF*08].

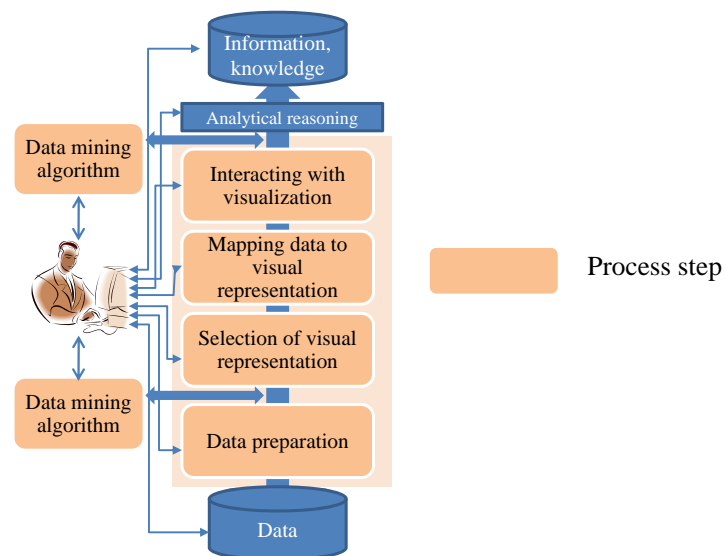


Figure 2.3.: Visual data mining process according to Simoff et al. [SBM08].

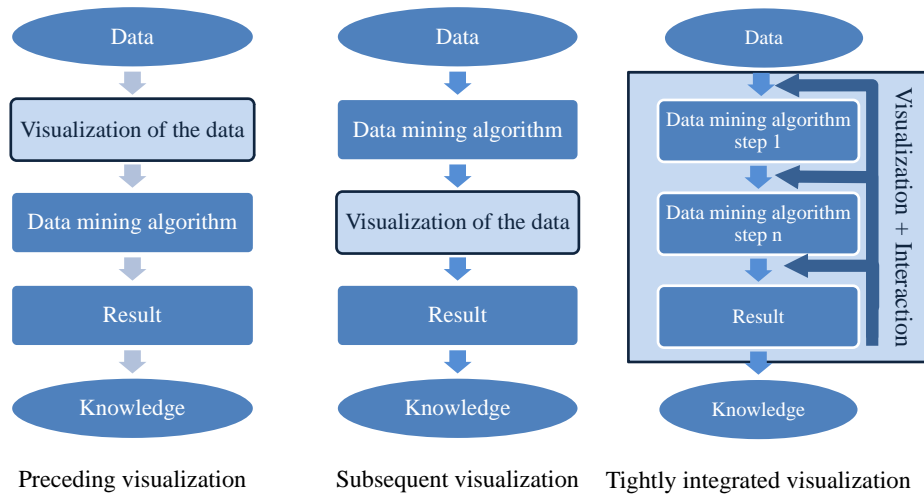


Figure 2.4.: Modes of visualization integration in data mining by Keim et al. [KW04].

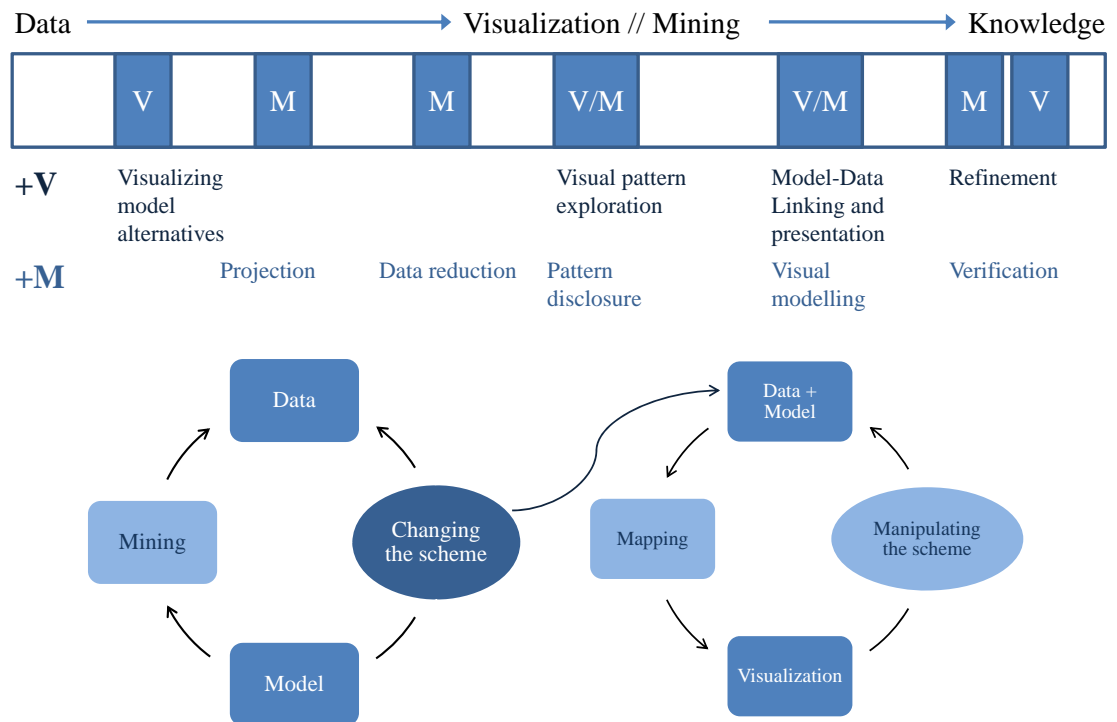


Figure 2.5.: Visual data mining according to Bertini et al. [BL09]. Top: Benefits from visualization and data mining in knowledge discovery. Bottom: Feedback loops between data, data models and visualization.

2.2.4. Visual Analytics Research Trends and Challenges

Visual Analytics Research Trends When looking at relevant works in Visual Analytics (based on publications from the main Visual Analytics conferences (e.g., IEEE Information Visualization Conference, IEEE Symposium on Visual Analytics Science and Technology, IS&T Visualization and Data Analysis, Eurographics/IEEE Symposium on Visualization, etc.) and main journals (e.g., Palgrave Information Visualization, IEEE Computer Graphics and Applications, IEEE Transactions on Computer Graphics and Visualization, etc.) in past couple of years), there are various topics of research which include

1. *design and evaluation of (collaborative) Visual Analytics systems*: papers on design of Visual Analytics systems solving particular analytical problems and use of collaboration in solving such problems (e.g., [CGK*07, Rob08, BMZ*06, BCB08, Kee06, KGS*08]). These papers are either focused strongly on usage of interactive visualization and data processing techniques for application benefit or on the collaborative interaction techniques,
2. *visual analysis of large complex data sets*: development of techniques for visual analysis of large amounts of complex data by combining information visualization techniques and data processing (in particular data mining) e.g., [BZL*08, AA07, IWSK07, EST07, NHM*07, SMER06, BB03, SSK07]). Please note that these works are very similar to papers mainly presented in visual data mining community,
3. *reasoning and “insight provenance”*: techniques for capturing the analytic process steps and the gained insight during the analysis; understanding of analytic process or reasoning systems, e.g., [FHRH08, GZ08, PBB*08, CYR09]. These type of works relate mainly to the knowledge management area,
4. *miscellaneous*: papers from various areas which are of direct interest to visual analysis of large data. These include, for example, efficient data management for visualization and analysis of large data sets, e.g., [CXGH08, TG07], theories [Che08], usability [Sch06].

There are no sharp borders between these topics, for example various techniques proposed in (2) are used for development of Visual Analytics systems (1) and supported by efficient data management (4).

In the thesis, we concentrate mainly on the first and second type. They focus on efficient combination of interaction, visualization and data processing techniques for efficient analysis of large complex data (for a specific application domain). Therefore, in the following text and the next sections we deal mainly with interactive information visualization, data mining and in particular their interplay.

Visual Analytics Challenges Owing to its short history, Visual Analytics has still a long way ahead, many challenges are still open. Four main works [KMS*08], [TC05], [TC06] and [LK07] have recently tried to summarize the research challenges in the area. These include

- facilitation of understanding of massive data sets also for data streams (i.e., scalability challenge),
- provision of frameworks for visual analysis high dimensional and time-dependent data (i.e., complexity challenge),
- inclusion of effective data transformation, data filtering and data mining for extracting and representing relevant content in the data,
- provision of platform independent visual analysis systems,
- understanding and integrating of analytic reasoning with semantics and tracking of analytic process,
- representation of data quality and uncertainty,
- support of interdisciplinary collaboration,

- evaluation of usability and user acceptability of the techniques including development of the methodologies,
- development of new interaction techniques and theories of interaction.

The application challenges in Visual Analytics include business, physics and astronomy, environmental monitoring, disaster and emergency management, security and software analysis areas [[KMS*08](#)].

2.3. Information Visualization

Information visualization (InfoVis) is the communication of abstract data through the use of interactive visual interfaces.
Keim et al. [KMSZ06]

Information visualization is, together with scientific visualization, part of computer graphics, visualization in particular. Information visualization deals with abstract data (data with no inherent spatial structure therefore not allowing for direct mapping to a geometry). The abstract data can be structured (e.g. time series, multi-dimensional vectors, graphs) or non-structured (e.g., text). Scientific visualization concentrates on physically defined data (e.g., with geographic location (geo-visualization) or with specific geometry in space (e.g., medical visualization, flow visualization).

Information visualization is, generally speaking, a graphical representation of data for supporting decision making. Models of information visualization have been presented in [CMS99, vW05]. Card et al. [CMS99] present a model, which concentrates on the technical process of visualization from the data to the resulting views with user interaction. Van Wijk [vW05] focuses on the process of gaining new knowledge from the data using visualization.

The choice of visualization is based on the breadth of the collection of visualization techniques and the purpose of the visualization. The effectivity of visualization depends on its design, the user experience and her perception and cognition abilities [vW05]. The design includes the choice of combination of visual mapping of the data, the representation of data structures, data pre-processing and interaction techniques. Information visualization design should be in accordance with the principles of perceptual and cognitive theory in order to be effective. In addition, interaction design and suitable data pre-processing together with the visualization technique play important role for suitability of visualization for data analysis. Please note that interaction and data preprocessing techniques are discussed in the following sections.

The main goal of designing information visualizations should be to present the data in an easily understandable way which supports interactive solving of analytical tasks. The main challenges thereby are complexity (support of multiple data types) and scalability (supporting large data sets). Specialized visualization, interaction and data processing techniques are employed to cope with these challenges. Please note that these challenges as well as techniques are common to Visual Analytics (see Section 2.2), thereby forming the core of many Visual Analytics methods.

General overviews of information visualization techniques, perception principles and interaction techniques in information visualization can be found in [CMS99, Spe07, War00]. Specifically in the context of visual data mining, several overviews of information visualization techniques and data mining techniques were presented in [FdOL03, Kei02, KMS02, KW04, SBM08].

In the following, we present taxonomies of visualization techniques and a new definition of data type space and then give an overview of representative types of visualization techniques according to data types. Please note that visualization techniques can be understood in a broad sense as including also specific data pre-processing (such as clustering) and interaction techniques which are parts of the interactive visualization design (see Section 2.4 and 2.5.4 for more details). In the following sections, we use the narrower definition excluding these two parts.

2.3.1. Taxonomy of Information Visualization Techniques

Visual Analytics methods are based on appropriate usage of available information visualization techniques. Therefore, in this subsection we present classifications of available techniques. We first give an overview of relevant taxonomies and then present our unification of the methodologies.

2.3.1.1. Relevant Taxonomies

Classifications of information visualization techniques have been proposed by several authors. We introduce them in chronological order.

One of the oldest and mostly cited taxonomies was presented by Shneiderman in 1996 – “Task by Data Type” [Shn96]. It identified seven types of data and seven types of analytic tasks and connected them in information visualization context. The data types considered are 1-, 2-, 3-dimensional data, temporal and multi-dimensional data, and hierarchy and network data. The tasks include overview, zoom, filter, details-on-demand, relate, history, and extract. These data and task types were a basis for further taxonomies (e.g., [Nor98, Oli99, CMS99]). Chi [Chi00] presented a taxonomy using the so-called “Data State Reference Model” which is based not only on data types but also on the visualization processing steps. It discriminates four data stages (from value to view), three data transformation types and four visualization pipeline steps. Keim et al. [Kei02] categorizes information visualization techniques according to data type used (1D, 2D, manyD, text, graphs etc.), technique employed (standard 2D or 3D display, iconic display, pixel display etc.) and interaction and distortion techniques used (standard, projection, distortion, zooming, linking and brushing, etc.). Tory et al. [TM02, TM04] categorized techniques basing on the models of data not according to data attributes. The basic idea is that visualization is used for interpretation and thus there the user has a model of the data and the visualization serves a specific purpose of the analysis.

Maybe the most comprehensible taxonomy for information visualization design unifying several taxonomies was proposed by Pfizner et al. [PHP03]. It covers the following dimensions: data, task, interactivity skill and context (information visualization design aspects) as well as input and output hardware, the software tools and human perceptual abilities. Data dimension discriminates, in accordance with Bertin [Ber81], between data relations (linear, circular, tree, graph, lattice) and data types (object, attribute, meta-data). User centered dimensions are task type (based on Shneiderman’s typology [Shn96]), interaction (based on Tweedie [Twe97]), user expertise (novice to expert users) and usage context (user’s intent and needs, the interaction history, user’s skills, output device). The taxonomy defines visualization types according to Bertin [Ber81], (static to animated, textual to graphic, variables of image (plane, size, value, relation) and differential variables (texture, color, orientation, ...) which stand in the middle of the whole design space. We use this taxonomy as reference in the following.

2.3.1.2. A Unified Taxonomy of Visualization Techniques

In the following, we present our unified categorization. We combine the previous categorizations in order to cover a broad range of visualization types which were not covered by single classification presented above. We employ the following categories/dimensions (based on the previous works, in particular on [PHP03]):

- *data type*: the type of input data to be visualized. For more details see the following section,
- *analytic purpose (task)*: the purpose, which should be supported by the visualization, or purpose of the visualization (e.g., [Shn96]),
- *output device*: the output device used for visual presentation (e.g. mobile phones [OKK04], large (tiled) displays, head-mounted displays (e.g., [JA07]), tables [Ise07], sonification [Cia04, NB02a, NB02b], etc.),

- *rendering dimensionality*: The dimensions employed in the rendering (2D e.g., [BN01]), 2.5D (e.g., [DE02], [SH05]), 3D (e.g., [SH05]),
- *number of views*: number of views in which data is displayed (e.g., one view or multiple linked views [DCCW08, PKH04, CGK*07], matrix views [EDF08]),
- *visualization metaphors (= visual form)*: type of visualization metaphors used (e.g., icons, pixel-based, node-links or matrices for graphs, etc.),
- *dynamics*: whether the data presentation is static or dynamic. For example, visualization of 2D time series can either be static showing point trajectories or dynamic showing animation of point position changes [TK07],
- *application dependency*: The dependency of the employed technique on the application domain (and thereby specific data semantics). This may relate also to the analytical task however is more oriented to specific types of semantic meaning of the data represented by visual representations commonly used (and therefore also interpreted) by the domain experts. For example, there are specific techniques for displaying stock market data (open, close, volume) as in [BCLC97, Rob03], or adaptation of treemaps to portfolio analysis [JT92], specific visualization of velocities and forces [DBPBS07],
- *level of expertise*: whether the technique is mainly for expert or novice users. This property is difficult to determine as previous user experience with visualization techniques is needed to assess this property,
- *miscellaneous*: specialized techniques such as ambient visualization (e.g., [SE04] and [SE05]), multi-modal techniques (combination with sonification [Cia04]).

The information visualization design can be seen as a combination of specific parameters along the above-mentioned dimensions. According to the goals of the thesis, we will discuss in more detail techniques according to data type specifying which other parameters they use. Please note that in the following we concentrate on techniques which are suitable of standard displays (single screens with standard PCs) and disregard miscellaneous visualization techniques (such as ambient visualization or sonification). In the following, unless stated otherwise, the techniques use static 2D visualization.

2.3.2. A New Definition of Data Type Space

The commonly used taxonomy of information visualization techniques [Kei02, Shn96, CMS99, War00] uses the following data types:

- 1-, 2-, 3-dimensional data, where the term dimension means the attribute cardinality,
- temporal data, i.e., time-dependent or dynamic data, data with time dimension cardinality greater than one,
- multi-dimensional data, usually meaning data with more than three dimensions,
- tree, referring to specific types of graphs with no cycles,
- network, or graph in general,
- text.

They often discriminate between number of independent and independent variables. Moreover, they differentiate between categoric, nominal and continuous variables. This classification of data types is simple, however has several drawbacks. The main disadvantage is the imprecise specification of the data type in particular for complex data. For example, in case of time-dependent networks, it is not clear whether they should be included in networks or in temporal data type. Additionally, they disregard uncertainties in the data as a specific aspect. These aspects are addressed only generally without focus on specificities of these dimensions by the taxonomies

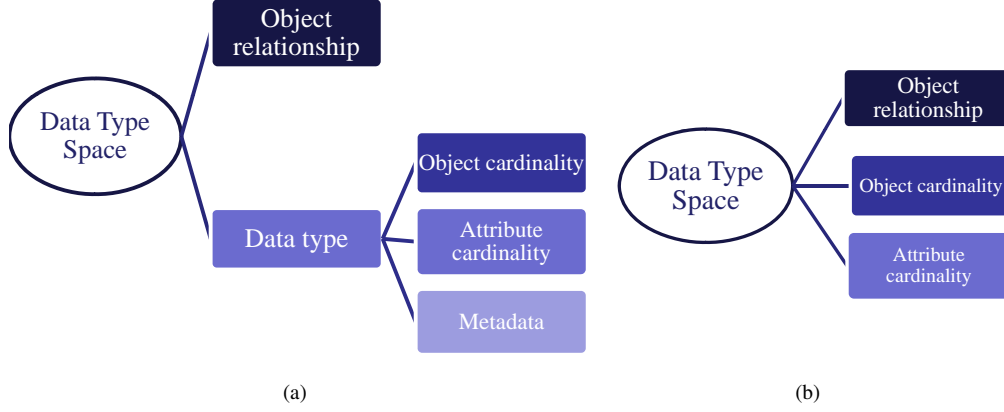


Figure 2.6.: Data type space definition according to Pfizner et al. [PHP03] (a) and Keim et al. [KMS02] (b).

presented below (see Subsection 2.3.2.1). We use these data type space definitions as a reference for our modified definition of the data type space presented in Subsection 2.3.2.2.

2.3.2.1. Related Data Type Spaces

In relevant studies, two main data type space definitions DS are mentioned: a) by Pfizner et al. [PHP03] and b) Keim et al. [KMS02]. We present and compare them in the following.

Firstly, Pfizner et al. [PHP03] differentiates between so called “data types” D_T and data relations R . The data types have three dimensions: number of data objects $|O|$ (i.e. number of items or object/item cardinality), number of data attributes $|A|$ and metadata M (see Figure 2.6 (a)). The object cardinality specification includes a single item, a set of items, or the whole set and refers thereby to selections of the original (input) data set. Data relationships consist of linear (rectilinear), circular, ordered tree (ordered pattern), un-ordered graph (pattern) and lattice (stereogram).

$$DS = D_T \times R, D_T = |O| \times |A| \times M \quad (2.1)$$

Secondly, a similar distinction between data items (called “information objects”), data attributes and relations between objects (called “information structure”) was presented in [KMS02]. They define data type space (“information space (DS)”) as the combination of attribute set A , information set (set of information objects) O and their relation R . According to the authors, this allows for modeling complex information spaces (see Figure 2.6 (b)).

$$DS = |A| \times |O| \times R \quad (2.2)$$

When looking closely at both definitions, an extensive similarity becomes apparent. Both definitions use data relation, object cardinality and attribute cardinality as dimensions of the data type space. Pfizner [PHP03] however includes also metadata and grouping of object (data type) attributes. Both taxonomies do not explicitly use the time-dimension and uncertainty of the data which, according to our view, is a major drawback for proper characterization of complex data types such as time-dependent multi-dimensional graphs. We use these two

definitions as a basis for our extended taxonomy presented in the next subsection which includes these two dimensions and further specifies the above mentioned dimensions.

2.3.2.2. Modified Data Type Space

The main dimensions of the extended data type space (i.e. “information space”) (DS), presented in our work, are data objects (O), time (T) and uncertainty (U):

$$DS = O \times T \times U, O = |O| \times A \times R, \quad (2.3)$$

In the previous works, time and uncertainty were not treated as separate dimensions. Data objects, time and uncertainty dimensions are further specified (see Figure 2.7 where our modifications are highlighted in red colors). Data objects are specified by their cardinality $|O|$ (size of data set), their attributes A and relationship R between them. Time and uncertainty are also further specified. These two dimensions can be seen as attributes of Keim’s data type space definition [KMS02], however, we think that these dimensions have specific meanings for visualization. Therefore we treat them separately. As an extension to the previous taxonomies, we specify the data type space dimensions in more detail. Our enhanced dimension specifications are derived from taxonomies for time [ABM*07], for relationship [PHP03], for attribute types [War00] and for uncertainty [THM*05, OM02]. Each input dataset can be characterized as a point in this multi-dimensional data type space.

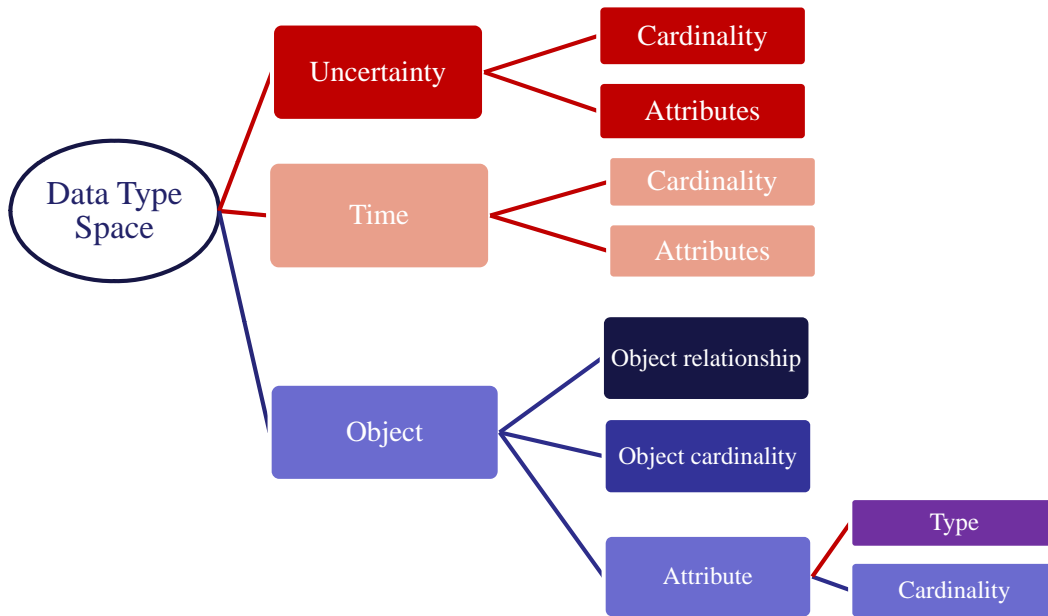


Figure 2.7.: A new data type space definition including also separate uncertainty and time dimensions.

In our view, the input data consists usually of one or more *data objects* (also called data point, data item, entities, etc.). With an increasing number of entities, scalability issues in information visualization arise. Each entity is described by 1, 2, 3, or more *attributes* (also called variables, features, dimensions). Each of the attributes can be cardinal (ordinal or nominal) or continuous. Note that one object may have multiple attributes and each

attribute can have different type, therefore the attribute type can be also mixed. There can be a *relationship* between the entities – either grouping (e.g. as a result of clustering, or categorization of the data), tree including hierarchic (e.g., as a result of hierarchic clustering or as a natural hierarchy), a general graph (general relations between entities) or compound graph (hierarchic and generic relations simultaneously). In case of compound graphs, the hierarchical and generic relationships exist within one graph at the same time. In social networks, for example, persons in an organization can be in a subordination (hierarchic) relationship and at the same time in a friendship (generic) relationship. Compound graphs can be also be created by successive aggregation (or clustering) of graph vertices.

The *time-dimension* of the data shows whether and how often the data objects, their attributes, relationships, types and locations change. In general, static and dynamic data are differentiated. For dynamic data, the attributes of time or events in time can be specified, e.g. frequency, periodicity of recurrence of events. Please note that the terminology for periodicity was altered with regard to Aigner et al. [ABM*07], the cyclic relationship we call seasonal (events in time recur regular time steps e.g., each December). We extend periodicity with cyclical events that occur on irregular basis e.g. after 5, 6, or 7 years. This distinction between cyclical and seasonal events in time is taken over from time series analysis domain often used in economic and financial applications.

In addition, attributes on *data quality* for each data point, for each variable, for each time-step may be available. The data quality information is often not available, in this case this dimension is disregarded in the analysis process. Note that the data quality information can be quantitative (how much uncertainty, error bands, error distribution) or qualitative (yes,no, the amount unknown). Also there are many types of data quality that can be regarded such as data accuracy, data provenance, data actuality simultaneously.

2.3.3. Overview of Information Visualization Techniques According to the New Definition of Data Type Space

In this section, we present an overview of information visualization techniques, which are relevant for Visual Analytics of abstract structured data based on the data type space definition presented above¹. Owing to a high number of alterations of the techniques, only the main ones are presented within each category.

We divide the techniques into groups as shown below. It enables a simplification of presentation of techniques according to the multi-dimensional data type space definition presented above. The first group concentrates on visualization of uncertainty, the second on visualization of static data and the third one on visualization of dynamic data. Note that the latter two sections disregard uncertainty dimension of the data (if available). Within the groups we further differentiate techniques according to object relationship type and then other object dimensions (object attribute type and dimensionality, number of objects). Please note that other types of groupings would be possible as well. Sometimes proper categorization of techniques is difficult as some techniques may be applicable to multiple types of data or, in particular, datasets of various sizes.

1. Visualization of data without uncertainty (disregarding uncertainty)

a) Visualization of static data (see Subsection 2.3.3.1)

- disregarding relationship between data objects,
- grouping,
- tree, including hierarchy,
- graph,
- compound graph.

¹Data type is one of the most important dimensions of the unified classification of visualization techniques presented in Section 2.3.1.2.

b) Visualization of dynamic (time-dependent) data (see Subsection 2.3.3.2)

- disregarding relationship between data objects,
- grouping,
- tree, including hierarchy,
- graph,
- compound graph.

2. Visualization of data with uncertainty (see Subsection 2.3.3.3).

2.3.3.1. Visualization of Static Data without Uncertainty

In the following, we present techniques for visualization of static data (or techniques that disregard the time dimension of the data, if available). The visualization of static data (sometimes called cross-sectional) has received much attention in the information visualization community. We present the techniques for visualizing data in 2D and 3D grouped by relationship between the data objects and, if applicable, the number of objects and the number of their attributes.

Techniques disregarding relationship between data objects *Two or three dimensional data* (data items with two or three attributes) can be shown in scatterplots encoding the data attributes by position in 2D or 3D space. Further data dimensions in scatterplots, if applicable, can be displayed using an appropriate visual metaphor (e.g., glyphs or icons). In case of cardinal attributes, jitter can be used to overcome over-plotting of data on the screen. For large data sets, overplotting can be also solved using transparency or point coloring by the number of data items or using pixel bar charts instead [KHL*01]. Pixel bar charts combine bar charts and scatterplots. They place the data items on individual pixels within the bars thereby overcoming the overplotting.

Datasets with items having *many attributes* can be visualized in several ways depending on the type of data attributes.

1. *Data with continuous values*: Scatterplot matrices [EDF08] show all pairs of data dimensions in a scatter plot view. They allow for spotting correlations between two dimensions, however are not suitable for multivariate analysis. For larger number of dimensions, scalability issues occur. Parallel coordinates [EN75] and radial plots [HM95, EST07] show all data dimensions in one view. Parallel coordinates place the data dimensions on vertical lines. Radial plots use a placement of axes in a circle crossing in the middle. The angle between the axes is constant. Data items are shown as lines connecting points (data values) on the axes. The advantage of these techniques, is the possibility to spot dependencies among several variables, however for large data sets they suffer from overplotting. Several techniques have been therefore proposed for clearer display of the data in such settings. Sometimes, data projection step is used as a pre-processing and then the reduced-dimensional data set is presented in a scatterplot. Projection views [FB94] follow this idea, while combining projection with sections (a kind of brushing).
2. *Data with categorical attributes*: Parallel Sets [BKH05] are a technique similar to parallel coordinates that can be used for categorical attributes. The dimensions are ordered in parallel axes and rectangular areas connect the data sets. This view shows the relationships among several variables. Mosaic Plots allow for analyzing relationships between categorical variables. The data is shown by splitting of a rectangle in alternative horizontal and vertical way for each variable, where the size of the rectangle represents the size of this set. KVMap [May07] follows a similar approach, however the splitting is uniformly sized and the color is used for showing the variable correlations.

3. *Data with mixed (or any kind of) attributes*: For showing data sets with multiple mixed variables, Table Lens [RC94, JTS08], icons (e.g., Chernoff faces [EN75]), pixel-based (circle segments [AKK96] or recursive pattern [KAK95]) can be used. Table lens are tables with unequally sized cells, with larger cells in the focus area. Depending on the type of variable and focus, the cells can display the numeric values, be color-coded, show bar charts or other visual representations of the data. This flexible view on the data allows for analysis of large multivariate data sets. Chernoff faces use iconographic representation of the data, where each variable is mapped to one visual attribute (e.g., size of eyes, hair, nose, etc.) In this way, multiple variables can be intuitively displayed, however for larger data sets, a comparison of the data may be difficult. Circle segments is a pixel-based technique, where the variables are represented as circle segments and individual items are arranged within the segments from the circle center. The coloring of the circle segment parts shows the data values.

For large data sets of many dimensions, pixel bar charts or pixel matrix displays are possible options [KHL*01, HDKS07] (see above).

Techniques focusing on data object groupings For data with continuous two or three dimensional attributes, techniques visualizing point clouds (groups of points) are relevant. These include hulls and distance based techniques. Hulls represent the point clouds with solid shapes, using various geometric constructs such as minimum bounding discs, boxes, and convex hulls [SP07]. The hulls represent the data in an abstract way showing their shape and distribution. These constructs may still lead to strong overplotting of the data, in particular for large data set with many data groups. Moreover, some of the shapes do not clearly represent the data shape. Therefore, compact enclosing shapes were introduced [SBTK08, CPC09]. Distance fields allow representation of point sets by smooth formation of visual areas by using appropriate transfer functions [KTSZ08]. By interactive varying of the transfer function, various shapes and data characteristics can be visually inspected.

Techniques focusing on tree (including hierarchic) structure of data objects These techniques can be divided into three main groups: space filling, node-link and mixed. There have been several studies comparing the different ways of tree visualization, in particular hierarchy visualization [BN01, AK07, Kob04, Sta00, vHvW02]. In general, it is difficult to unify these results as they differ significantly. Recently, it has been found that the effectivity of the respective technique largely depends not only on the task to be solved, but also on the formulation of the task assignment, i.e., if it reflects a containment or a levels metaphor [ZK08].

- *Space filling techniques*: These are mainly applied to rooted trees. They use the spatial position of the nodes (such as closeness or enclosure) to represent the hierarchic structure of the graph. Moreover, they try to use the full area of the display to present the graph. They are mainly used to visualize the hierarchic partitioning of the set of all data items into partitions, e.g., when considering the set of files in a standard file system. The size of the nodes is encoded by the area size of the displayed items. Additionally, color and height can represent additional data attributes. In case more complex additional information needs to be displayed, specialized data presentations can be placed in the child nodes such as icons, parallel coordinate diagrams, etc. Space-filling techniques can be categorized by the placement strategy employed into enclosure (e.g., treemaps [Shn92]), adjacency (e.g., Sunburst [SZ00]) and crossing (e.g., Beamtrees [vHvW02]).
- *Node-link techniques*: These approaches use links between items to depict their relationship. Layout algorithms controlled by optimization criteria or layout heuristics calculate a layout for the positions of the nodes. The method by design typically leaves significant background space empty and thereby may encounter scalability problems when applied to larger graphs. Many layout algorithms have been proposed to date in the graph drawing community. They include radial or balloon layouts in 2D [HMM00], Cone trees [RMC91] in 3D, point based trees [SSH09], nature inspired Phyllo trees [NCA06], or Hyperbolic lay-

outs [Mun97, AH98]. For the visualization of node attributes, specialized techniques for multi-dimensional data visualization such as glyphs, radial or parallel plots can be used.

- *Combined node-link and space filling:* These approaches combine node-link diagrams with treemaps. In these, a part of the hierarchy is displayed in an enclosing (treemap) mode, and the rest as a node-link diagram. They present the data in a flexible space-efficient way while still clearly presenting the data structure and emphasizing the content. The most prominent representative are “elastic hierarchies” [ZMC05].

An alternative approach for very large data sets are hierarchic pixel bar charts [KHD02].

Techniques focusing on general graph structure of data objects Graph visualization techniques can be classified according to the visual metaphor used into node-link, matrix or combined representation. A comparison of node-link and matrix techniques [GFC04] shows that node-link diagrams are more intuitive, compact, are better suited for path following tasks. Matrix data display do not suffer from overlapping problems, therefore can represent also dense graphs. Both techniques suffer from scalability in limited display spaces. Both graph layout and matrix node ordering influence their effectiveness.

- *Node-link diagrams:* The data is displayed using nodes and links between them. Graph drawing and graph layout are large research areas, where many approaches have been developed (see [DBETT99, DPS02] for an overview). The layout techniques can be divided into force-based layouts, constraint-based layouts, multi-scale approaches, layered layouts, and further approaches, which are explained in [vLKS*10]. Moreover, the related work part in [AAM07, MM08] and the comparison of layouts in [HJ07] can be used as a reference.

In addition to specific layouts, occlusion and readability of the display can be improved by edge-bundling [CZQ*08, Hol06] and the removal of node overlap [GH09, IAG*09].

Drawing of node-link diagrams also includes a suitable design of edge and node drawing primitives. For directed graphs, the representation of edge directions is of importance. There are multiple design possibilities including usage of arrows, color transitions (from color A to color B), thickness transitions (from thick to narrow), curves, and animated textures [HvW09, TK08, BBG*09]. These options may also be combined. For edge weight, coloring of edges or edge thickness can be employed. For the visualization of node attributes, a visualization of multivariate data items (e.g., glyphs or radial plots) is employed.

- *Matrix:* These techniques visualize the adjacency matrix of a given graph, where edge attributes are encoded in the matrix cells. It can display both directed and undirected graphs, where the latter leads to a symmetric matrix. In a matrix visualization, the ordering of rows/columns plays an important role. A proper reordering can reveal clusters in the graph and other patterns. Relevant techniques are discussed in [EDG*08, HF06].
- *Combination of matrix and node-link approach:* Techniques using a combination of the two previous approaches aim at overcoming their limitations by focusing on their strengths. Three main approaches exist: multiple synchronized views (linking the matrix and node-link representation [HF06]), Matlink [HF07b] (enhancing matrix visualization with links at the border of the matrix) and NodeTriX [HFM07] (combining both representations in one view, where node-link diagrams display the overall graph structure of the network, and adjacency matrices show communities).

Techniques focusing on compound graph structure of data objects Literature on visualization of graphs with hierarchic structure is relatively rare. We identify three main approaches: node-link diagrams, treemap-based and matrix with links.

- *Node-link graph visualization techniques:* These use node-link diagrams for the lowest hierarchy level and then use “bubbles” (enclosures) for various hierarchy levels. Examples include TugGraph [AMA09] and GrouseFlocks [AMA08]. The advantage of this method is its intuitiveness. However, for large graphs with many links, this view gets easily overcrowded. This problem can be partially solved by edge bundling [Hol06] or by showing only links between merged nodes.
- *Treemap-based:* A Treemap visualization of the node hierarchy uses overlaid links between nodes [FWD*03]. This approach may suffer from strong overplotting in case of many links between nodes of the hierarchy. Therefore, edge bundling is advised to improve the readability of the display [Hol06]. Similarly, also one-dimensional Treemaps with links between nodes, so called ArcTrees [BDJ05] can be employed, but these do not scale well for large hierarchies.
- *Matrix view with links:* These visualizations combine the generic node relationship visualization with a tree-based visualization of the hierarchic node relationships. This is an analogy to MatLink [HF07b]. This view is very clear, however, it may be difficult to understand the compound relationships between nodes.

2.3.3.2. Visualization of Dynamic Data without Uncertainty

There are two main approaches to the visual display of the time changes on graph elements: using animation, and using static displays. Animated displays usually employ or enhance static visualization techniques such as presented in Section 2.3.3.1. Animation is a natural way of conveying the change of the data over time. However, its effectiveness is limited by human perception capabilities. Usually, users are able to recognize and remember larger changes in the data. The static view is preferred for more detailed analysis of data changes. Static views which also incorporate the time-dimension of the data are more complex.

For data with tree, graph and compound graph structure, we categorize the visualization techniques according to the type of data changes captured into those that affect only *data attributes*, and those that affect also *data relationships*.

Techniques disregarding relationship between data objects We divide the techniques in the following according to the number of data attributes of data objects.

- *One dimensional attributes:* For one dimensional time series, classic line charts can be used. Line charts are intuitive and well display the data movements. However, they suffer from overplotting in case of large data sets. Therefore, in line charts with many data objects, histograms [RW04] use highlighting frequencies of data attributes. Long data series can be also explored using interaction techniques such as semantic zooming and brushing [HS04]. Line charts do not explicitly reveal periodic movements (seasonality) in the data. Time spirals [WAM01] are suitable for this purpose. They show the data values along circular segments in a spiral way (from center to the outside) using color-coding. The number of segments should match the periodicity of the data.
- *Two or three dimensional (continuous) attributes:* Two- and three-dimensional dynamic data can be displayed using animated scatterplots or trajectories [TK07], [Gap]. Animation reveals the main data dynamics and trajectories show detailed data movements. Showing trajectories of all data items in one view may lead to strong overplotting and thereby to unclear data views. Data filtering or multiple data views can be used in this case.
- *Multi-dimensional (continuous) attributes:* For displaying multi-dimensional time-varying data, parallel coordinates and radial plots – Parallel Glyphs [FCI05] can be used. Parallel Glyphs integrate parallel coordinates and radial plots. The radial plots show the multi-variate data in each time step. They are situated in parallel planes to display the time-dimension of the data. The data-values in the data points are connected

using lines. TimeWheel [TAS04] is an axis-based visualization (similar to parallel coordinates or radial plots), where one axis represents the time and the other the data variables. Several axis positions have been proposed, e.g., where the time is in the middle of a circle and the variable axes are positioned across circle segments. In case that the attributes add up, Narratives or ThemeRiver visualizations [FHRH08, HHWN02] can be used. ThemeRiver shows the data using a river flow analogy. The data variables are represented as colored streams in a river. The stream width (and inherently the river width) represents the values of the data in a time point. Pixel based techniques for large datasets were presented by Hao et al. [HDK08]. The data display is similar to pixel bar charts [KHL*01] with additional time-dimension.

Techniques focusing on data object groupings As an extension to the techniques of for visualization of data groupings in static case, animation or trajectories of hulls can be used for data sets with two or three dimensional data attributes [vLBR09]. The animation is better suited for revealing major data developments and trajectories for detailed view on the data changes. Trajectories, however, may suffer from strong overplotting, especially for large datasets.

Techniques focusing on tree (including hierarchic) structure of data objects For the visualization of the data with only data attribute changes, either treemaps with time series in the leaf nodes [SKM06, DHKS05] or the so called Timeline Trees [BBD08] can be used. The treemap representation directly shows the hierarchic structure and time-variation in one combined view. Timeline Trees show the hierarchy on one side and the time sequences on the other side of the view.

For visualization of dynamic data with structural changes, animated views are used. In this respect, animated graphs can be employed in general. In particular, the layouts based on the Sugiyama approach [GBPD04] are suitable. Alternatively, animated treemaps [GF01, TS07] or icicle/circular plots [TS08] can be used. When choosing the graph layout, the layout stability needs to be taken into consideration. For example, in the treemap representations, the spiral layout [TS07] achieves a high continuity with high stability of the layout.

Techniques focusing on general graph structure of data objects For attribute changes only, techniques for visualization of static graphs can be combined with visualizations of individual time dependent data items (e.g., color charts [SLN05]). The advantage of this approach is the large number of the available graph layouts.

In case of structural changes, time-dependent graph layouts (animated graphs) need to be employed [KG06, DGK01]. In animated graph visualization, a stable graph layout, which changes minimally, is of essence. There is a difference between strategies for drawing graphs with known histories [KG06, DGK01] and those that need to be adjusted in real-time depending on new data streams [FT08].

Techniques focusing on compound graph structure of data objects There are only few techniques that visualize time-varying compound graphs. They employ either animation or static data representations. Static approaches include TimeArcTrees [GBD09] (a sequence of node-link diagrams with horizontal node alignment), TimeRadarTrees [BD08] (radial tree layouts for the hierarchy and a sequence of circle segments for representation of the temporal change of the structure). Animated approaches include specific layouts. Kumar et al. [KG06] present animated node-link diagram with transparent “bubbles” for the hierarchic grouping of nodes. Frishman and Tal [FT04] propose a layout where the groups of nodes are displayed using bounding boxes around the groups.

2.3.3.3. Visualization of Data with Uncertainty

Data uncertainty (quality) is not in the focus of the thesis, therefore we provide only a short overview of techniques used for presentation of data uncertainty. Extensive overviews of methods for visualizing error and uncertainty are presented in surveys [PWL97], [JS03], [MRH*05] and [GS06].

The available techniques include:

1. usage of free graphical variables: color, size, saturation of color, position, angle, clarity, fuzziness, transparency, edge crispness,
2. integration of additional graphical objects: uncertainty glyphs, labels, isosurfaces, textures,
3. usage of animation: speed, duration, blinking, motion blur,
4. interactive representation: e.g., clickable map, difference images, mouse over effects, magic lenses,
5. addressing other human senses: acoustic or haptic senses (e.g., sound or vibration).

A user study of the methods [KMB03] for spatial data has shown that the most useful techniques are blinking, and overlay. At the same time animation and saturation of color were deemed least useful.

The challenge of the uncertainty techniques is to support a combination of qualitative and quantitative uncertainty information for abstract data [GS05]. Although many techniques for multivariate data visualization exist, techniques for visualization of multivariate data uncertainty are still rare. They include the approach of Schmidt et al. [SCB*04] for parallel coordinates for environmental data, the visualization technique by Davis et al. [DK97] for multi-variate spatial uncertainty for geo-spatial data or multivariate visualization using glyphs for multivariate both cardinal and continuous uncertainty attributes by Tekušová et al. [TKSK08].

2.4. Interaction

Interaction between human and computer is at the heart of modern information visualization and for a single overriding reason: the enormous benefit that can accrue from being able to change one's view of a corpus of data. Usually that corpus is so large that no single all-inclusive view is likely to lead to insight. Those who wish to acquire insight must explore, interactively, subsets of that corpus to find their way towards the view that triggers an 'a ha!' experience.

Spence [Spe07]

In the following, we first present relevant categorizations of interaction techniques according to the fields of interest to Visual Analytics (information visualization, reasoning and data processing). We then present our unified taxonomy which alters and combines the described methodologies.

2.4.1. Relevant Taxonomies of Interaction in Information Visualization, Reasoning and Data Processing

Interaction in Visual Analytics includes several means of interaction differentiated according to the three relevant fields: information visualization, reasoning and data processing. The seminal study by [War00] extensively described interaction techniques and levels of interaction in information visualization. Recently, three papers dealing with the theory of interaction in information visualization [YKSJ07], reasoning and information visualization in sense of Visual Analytics [GZ08] and visual data mining with focus on visualization and data mining [BL09] have been published. The first paper provided an extensive survey of interaction techniques, tasks and operations and introduced a new taxonomy based on user intention. The characterization of analytic activity (reasoning) in Visual Analytics and the definition of a taxonomy of actions based on intentions was presented in the second paper. The third paper focused on the interaction with (combined) data mining and visualization. In the following we present the taxonomies of all the above-mentioned studies (including Ware [War00]) and try to compare them. Please note that the mentioned studies do not consider interaction on large displays or collaborative interaction, which is in accordance to the goals and assumptions of the thesis stated in the introduction of the thesis.

Information visualization: In Information Visualization, interaction plays a major role for gaining insight into the data via exploration and navigation and for overcoming the scalability and complexity problems – e.g., occlusion, screen limitations and fundamental human perception limitations.

Ware [War00] introduced three levels of interlocking feedback loops when interacting with visualizations (data manipulation, exploration and navigation as well as a problem solving loop). The data manipulation loop entails selecting and moving objects using eye-hand coordination. The exploration and navigation loop helps finding a way in the data space and thereby building a mental model of the data. In the problem solving loop, the analyst forms hypotheses about the data and refines them through an augmented visualization process. Interaction is commonly understood as the second feedback loop (level). However, the most recent articles [YKSJ07, GZ08] try to use user intention for classification purposes and thereby to expand the exploration and navigation techniques (i.e., classic interaction techniques).

Exploration and navigation in information visualization include

- *view navigation* = changing view control, e.g., panning, walk-through,
- *focus, context and scale* = moving between views on different scale including distortion techniques, rapid and semantic zooming, multiple windows, elision,
- *rapid interaction with the data* = being in direct contact with the data

- *change of data mapping* (e.g., mapping of data attributes to x and y axis, to shape, color),
- *change of data transformation for presentation* (e.g., changing of parameters of transfer function)
- *dynamic queries* (i.e., filter) = limit range of data that is visible,
- *brushing (and linking)* = highlighting subsets of data interactively (in multiple windows).

This simple but expressive taxonomy on an abstract level includes the main classic interaction techniques for data exploration and navigation. It does not distinguish clearly between these two types of actions. It is more centered on system functions, not on user intentions (i.e. it is a low-level taxonomy). This categorization is however in line with the Information Visualization reference model of Card et al. [CMS99] where interaction actions follow the information visualization pipeline. By contrast, user intentions are the basis for the following two taxonomies.

Yi et al. [YKSJ07] present an extensive overview of interaction taxonomies, tasks and dimensions for information visualization. They propose a new taxonomy based on user intentions, which includes seven types of intentions and the associated techniques. In the following, we try to match this taxonomy with the taxonomy of Ware [War00].

- *Select*: marks something as interesting. Corresponds to brushing in [War00].
- *Explore*: enables users to examine a different subset of data, includes, for example, panning and direct walk. Corresponds to view navigation in [War00].
- *Reconfigure*: provides users with different perspectives of the data by changing the spatial arrangement of representation. This includes sorting and rearranging columns, changing attributes assigned to x and y axes, reducing occlusion (e.g., rotation, jitter). This intention corresponds to rapid interaction with the data in [War00]. However it includes techniques for solving occlusion by view navigation, which does not seem intuitive.
- *Encode*: alters the fundamental visual representation of the data including visual appearance. Although there is no direct correspondence to one of the items, it entails attributes of changes in data transformations for representation and data mapping.
- *Abstract/Elaborate*: adjusts the level of abstraction of a data representation, e.g. details on demand, focus on data, tooltips and zooming. This intention corresponds to focus, context and scale in [War00].
- *Filter*: changes the set of data items presented, e.g., dynamic queries. This intention corresponds to dynamic queries in [War00].
- *Connect*: includes techniques used to a) highlight associations and b) show hidden data items that are relevant to a specific item e.g., multiple views that reveal items that are not directly shown (e.g., show children of a node in a graph). It does not have a direct correspondence in the taxonomy of [War00], but it is similar to explore (also according to Yi et al. [YKSJ07]).

In their paper, Yi et al. [YKSJ07] mention further techniques that were not categorized such as undo/redo, change system configuration, threshold highlighting (could be part of select), semantic zooming (fulfilling multiple intents). Many of these techniques may not have been classifiable owing to their proximity with elements of interaction belonging to reasoning and to data processing.

In general, from the strong correspondence of the two types of taxonomies (by Ware [War00] and Yi et al. [YKSJ07]), it can be seen that the user intentions and low level interaction techniques correspond to a very large extent when talking about interaction in information visualization.

Reasoning The taxonomies in information visualization focus on the manipulation of visualizations and therefore do not include further analytical (insight) elements such as annotation and change in view history, features

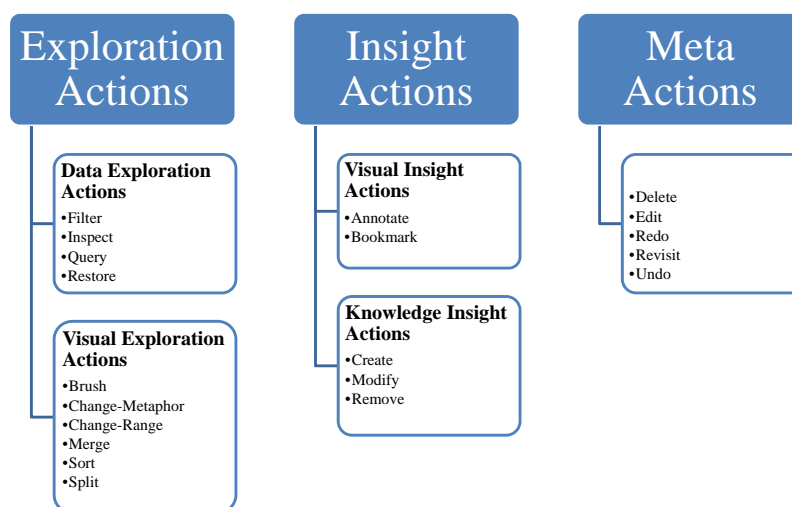


Figure 2.8.: The action taxonomy according to Gotz and Zhou [GZ08].

which are deemed relevant for user interaction from an analytic (reasoning) point of view. These are mentioned in the taxonomy for Visual Analytics concentrating on reasoning and visualization [GZ08].

Gotz and Zhou [GZ08] create a taxonomy of user activities in the analytic process with three main categories: exploration actions, insight actions, and meta actions (see Figure 2.8). Exploration actions are divided into data exploration (Filter, Inspect, Query, Restore) and visual exploration (Brush, Change-Metaphor, Change-Range, Zoom, Pan, Merge, Sort, Split); insight actions into visual (annotate, bookmark) or knowledge-based (remove, modify, create) and meta actions include the following types: redo, undo, revisit, delete. Taken together, they define four distinct intents: (1) data change, (2) visual change, (3) notes change, and (4) history change.

We see that the exploration actions roughly correspond to the interaction actions discussed above [War00, YKSJ07] with respect to information visualization. Moreover, with respect to analytic activity Gotz and Zhou enhance the types of interactions with insight and meta actions, which were mostly neglected before as they refer to the reasoning part of Visual Analytics. Both Yi [YKSJ07] and Gotz and Zhou [GZ08] use user intentions to characterize interactions. By contrast with Gotz and Zhou, Yi et al. do not regard insight provenance actions such as notes and history change, which seem more closely related to the reasoning element of Visual Analytics.

Data Processing All the above mentioned taxonomies do not consider interaction with data processing (data mining in particular) tools as presented in [BL09]. With regard to visual data mining tools, Bertini and Lalanne [BL09] discriminate between pre- and post model interventions to change the scheme or manipulating the scheme for both visualization and data mining. They thereby take a different approach to the categorization of interaction techniques. They look at whether the user action only changes the current state (tuning, change of parameters) or changes it completely (change of the scheme) Both means are presented in the following.

- *manipulating and tuning*: change parameters within the context of a given scheme,
 - in visualization: changing representation parameters (e.g., zooming, etc.),
 - in data mining: changing model parameters (e.g., changing distance function),

- *changing the scheme*: changing the data model or representation,
 - in visualization: changing the visual mapping or visual representation (see Ware [War00]),
 - in data mining: changing the data model (e.g., change from generation of rules to finding data clusters).

In summary, the notion of interaction with respect to Visual Analytics is broader than the scope of interaction in information visualization. In information visualization, interaction concentrates on the exploration and the navigation of the data space. In Visual Analytics, it also incorporates capturing user insights, tracking of analytic activity (including navigation in previous analytic actions) and interaction with data mining tools for creating data models.

2.4.2. A New Unified Taxonomy of Interaction in Visual Analytics

The outcome of the previous analysis suggested the definition of a new unified/modified taxonomy based on the above-discussed taxonomies merging the three areas related to interaction within Visual Analytics – visualization, reasoning and data processing (data mining) (see Figure 2.9). Within the analytical process, the unifying element is constituted by allowing for interaction to take place between all three areas, thereby resolving the previous separation. The integration of the three interaction areas enhances the flexibility and ease, when changing the analytical subtask/focus, without losing sight of the overarching analytical process. It also entails the seamless change from one part into another such as from algorithmic data processing to visual exploration of the data and analytic annotation of the found insights.

Especially in *integrated visual analytic systems* these seamless changes invoked by a single interaction step are a significant feature. For example, when starting a new clustering algorithm the results may directly be presented in a visual way and then can be further interactively explored in the view. Vice versa, changes of parameters in the view may directly invoke new algorithmic calculations. Thereby the closing of the analytical loop is achieved. This integration feature however makes categorization of interaction actions more difficult. A question stands out, whether such actions should be declared to processing actions or visualization actions. In our view, if an action changes algorithmic parameters and the integrated features automatically imply changes to the visualization than both actions should be considered. Each of them is assigned in the taxonomy. In particular talking about tracking of user actions, both the immediate and the implied actions should be tracked simultaneously or a combined way of tracking should be developed.

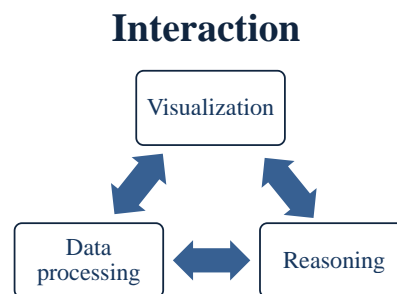


Figure 2.9.: Interaction in Visual Analytics as integrated interaction with visualization, data processing, and reasoning and among them.

The improved feature of the unified taxonomy presented here is the consistent labeling of the interaction techniques previously assigned to the three different areas, which were often categorized in isolation using different frameworks of reference.

We categorize user interaction according to the *action* that is taken by the user. This categorization is more suitable for dividing interaction techniques into categories than division according to user intention, as each action is supported by the employed technique. Note that the two categorization approaches are interrelated. A user intention can be achieved by several user actions or, vice versa, an action can suit several intentions.

Our unified approach is based on the method of Bertini et al. [BL09] for visual data mining. In our model, accordingly, each interaction area (Visualization (V), Reasoning (R) and Data Processing (DP)) includes two subcategories: changes in the data and changes in the respective representation (see Figure 2.10).

The changes in the data affect the presented/underlying data set and changes in representation refer to other forms of interaction. Changes in the data are divided into two subcategories: changes affecting the selection of the data set (in particular filtering) and changes to the data set introduced by the user (e.g., by editing the data or by annotating). The representation changes are divided (in line with Bertini [BL09]) into changes of representation parameters and changes of scheme. This categorization is in line with the Information Visualization reference model of Card et al. [CMS99]. Please note that these types of interaction are often closely connected. For example, data manipulation may automatically lead to changes of visual parameters (e.g., data filtering can influence the graph layout, or zooming can be combined with data filtering forming a type of semantic zooming).

In *visualization*, changes in representation include view changes (e.g., zooming, panning) and scheme changes include inter alia change of visualization type or visual mapping.

In the *reasoning* part, parameter change include, for example, undo/redo actions and scheme changes include inter alia change of analysis type often connected to a change of the sub-task performed.

In *data processing*, change of representation refers to changes in processing parameters and change of scheme refers to change of processing type (e.g., from clustering to dimension reduction).

The interaction means for visualization, data processing and reasoning as presented in [GZ08], [War00], [YKSJ07] and [BL09] can be assigned to the respective subcategories following the above mentioned categorization. For example, in visualization, data changes include filtering (dynamic queries), details on demand or data editing via direct manipulation. In visualization changes, scheme interaction includes changes to visualization mapping and type of visualization additionally parameter changes are in analogy to navigation interactions (panning, zooming, etc.). In reasoning, data changes refer to mainly tracking of the analysis process and insight actions [GZ08] such as editing of annotations by the analyst, the other category includes mainly meta actions [GZ08] such as undo, redo, revisit. Data processing interaction refers to the changes of data for the algorithmic processing via e.g., filtering or editing and changes to the algorithm parameters, changes of the type of processing algorithm or method.

It needs to be noted that assignment of interaction actions into the particular category on such an abstract level encompassing a broad scope of systems dealing with different data types, including a broad variety of functions is very difficult. Therefore this abstract categorization needs to be adapted to the specific examples.

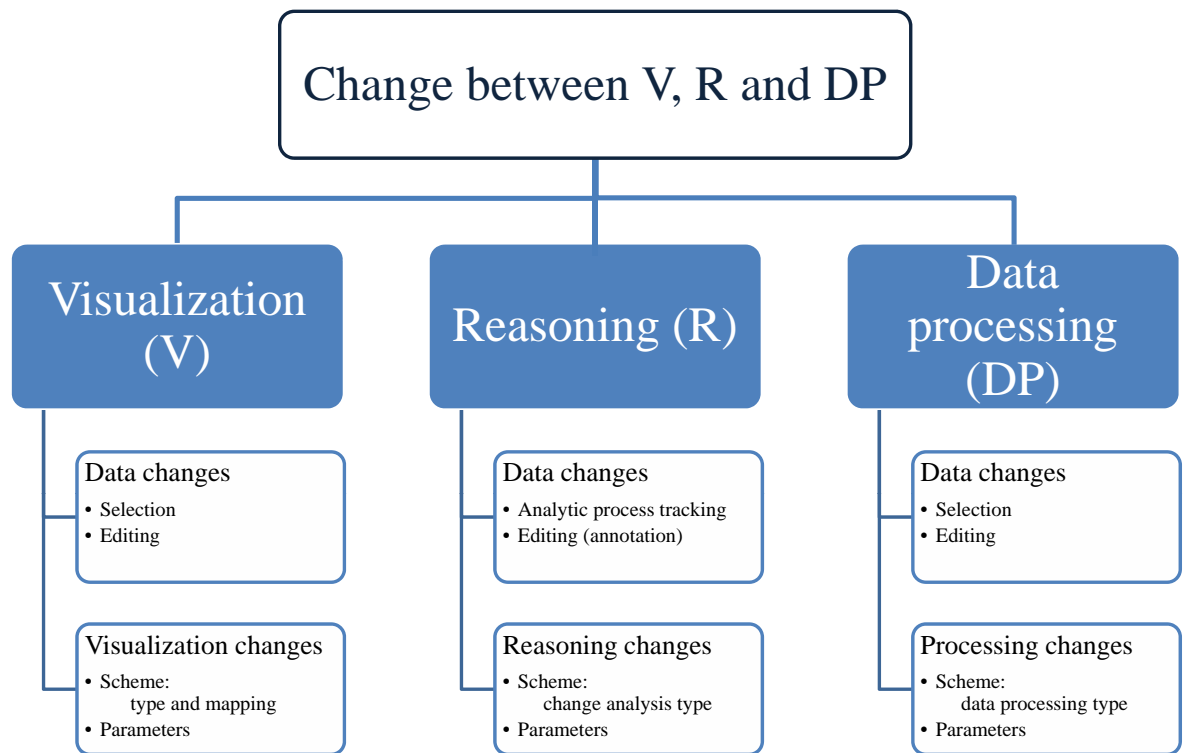


Figure 2.10.: A new unified interaction model showing two analogous interaction means with visualization(V), reasoning (R) and data processing (DP) as well as their interplay.

2.5. Data Processing

Computer-based data analysis (data processing) methods center mainly on two aspects: first, the automatic detection of interesting patterns in the data and the verification of user hypotheses, and second, the transformation of the data into new meaningful forms [TC06, KMS*08]. Data cleaning methods such as outlier correction or missing data replacement, while potentially included in this general definition are deemed not relevant for this thesis and thus disregarded going forward. Data processing methods are an inherent part of Visual Analytics and are commonly applied at different stages of the analytical process [KMS*08]. For example, these techniques are often used as a pre-processing step in data visualization (see also Subsection 2.5.4).

Data processing methods in the focus of the thesis include data transformation, data mining and application dependent techniques, which are described in the following. In the data mining section, the clustering technique is described in more detail and, in particular, self-organizing maps. These techniques are later used in our approach to the visual analysis of weighted directed graphs and two-dimensional time dependent data.

2.5.1. Data Transformations

The main goal of using data transformation techniques is to generate a reduced, more suitable, representation of the data set, which is smaller in volume or more suitable as input for further processing (in particular with regard to data type), but still produces similar analytical results [HK06].² Going forward, we present an overview of the main data transformation techniques. These techniques may also serve as pre-processing for other data processing techniques (e.g., data mining).

Reduction of number of data items includes, for example, clustering, aggregation and generalization as well as sampling and filtering [HK06].

- In *clustering* (see next section) a group of objects is replaced by one representative object based on similarity.
- In *aggregation and generalization*, a group of objects is represented by one object which entails other objects, e.g., using hierarchies (street-city-country) or other types of groupings (e.g., being part of a graph substructure). Usually the attributes of the new object summarize the attributes of the members of the group (e.g., averaging, summing up). This technique is often used in data visualization in order to reduce the number of objects that need to be rendered (e.g., in graph aggregation).
- In *sampling and filtering*, the techniques reduce the number of data items by choosing a subset of items from the original data set.
 - *Sampling* tries to select a part of the input data set that represents in an appropriate way the characteristics of the population data set (e.g., simple random sampling, systematic sampling, stratified sampling).
 - *Filtering* selects a subset of the data according to specific criteria, such as data attribute values (e.g., all companies from Germany).

Dimension Reduction Techniques (attribute cardinality reduction) reduce the number of data attributes, for example, by removing irrelevant attributes (also called attribute or feature selection) or by data projection

²If not referred otherwise, this section is based on [HK06].

into lower dimensional space. Various methods for the selection of relevant attributes have been proposed (see [LM07]).

Data projection techniques can be divided into linear and non-linear approaches.³ Linear projection methods use linear combination of original attributes to construct derived attributes. These techniques include inter alia principal component analysis (PCA), factor analysis (FA), independent component analysis (ICA). Their main advantage is simplicity. Their disadvantage is the constraint to linear transformations and thereby finding only linear structures in the data. Non-linear techniques do not restrict the calculation of derived attributes to linear combinations of the original attributes. The techniques include inter alia non-linear PCA, multi-dimensional scaling (MDS), and Sammon's Mapping (SM).

Summarization and discretization both reduce the number of data items. They either change the type of data attributes and/or reduce the number of attributes.

Summarization, for example, entails the calculation of summary statistics (average, mean, variance, quantiles etc.) for an input data set to get an overview of the data items in the set.

Discretization reduces the number of distinct values (e.g., from continuous into discrete space) for a given data set and groups the data items according to the discretized data values (e.g., histograms).

Transformation into another data type is often used for creating a suitable (and often simplified) data representation of the input dataset to be used in further analysis (e.g., as input into data mining or visualization algorithms).

Transformation techniques include, for example, feature extraction, data compression, summarization, discretization. *Feature extraction* is a data transformation in which input data objects are described by new (relevant) attributes. There exist a wide variety of feature extraction techniques specialized to particular data types (e.g., graphs, text, images). An overview of these techniques would go beyond the scope of the thesis. However we discuss specific techniques for feature extraction in graphs in Section 3.6.2 and two-dimensional time dependent data in Section 4.6.2.

Data compression creates a reduced representation of the data, whereby the input data can be reconstructed without any loss (e.g., wavelet transforms). This is the main difference to feature extraction methods, where the input data may not be reconstructed.

Miscellaneous techniques play mainly a supporting role. Examples are normalization, scaling, partitioning, calculation of data properties (e.g., in graphs number of roots).

2.5.2. Data Mining

Data mining, generally speaking, is the process of extracting hidden patterns from data algorithmically. Data mining methods include classification, regression, clustering, summarization (see Section 2.5.1), dependency modeling, change and deviation detection, association analysis, outlier detection [FPsS96, HK06].

- *Classification* concentrates on finding a set of modules/functions that describes and distinguishes data classes or concepts for the purpose of determining classes of unlabeled objects. More specifically, the output are categoric values.

³Attribute selection techniques can be regarded as linear orthogonal projection techniques.

- *Clustering* is an unsupervised method for the grouping of objects, which maximizes intra-group similarity and at the same time maximizes inter-group dissimilarity. In contrast to classification, objects to be clustered have unknown classes.
- *Association analysis* (also called association rule learning or mining) focuses on the search for interesting relationships among data items. Alternatively, it tries to discover association rules showing attribute value conditions that occur frequently together in a given data set.
- The *regression* methods model the data by approximating the given data (e.g., linear regression). In contrast to association rules, input and output data are continuous in nature.
- *Outlier analysis* tries to find outliers in the data. Outliers are such data items, which are extraordinary or which do not comply with the general behavior of the data.

As can be seen, there are many types of data mining methods. For the purposes of the thesis, we concentrate on clustering in the following.

2.5.2.1. Clustering

Clustering is an important data mining technique, which has its root in statistics. It has been applied in various areas. Clustering supports the examination of large amounts of data by revealing their structure in the data set and observing characteristics of groups of data [HK06]. It abstracts in an unsupervised manner data objects into a limited number of data groups (i.e., clusters). The clustering results can be used as input to various further applications.

The objective of clustering is to group together objects having a high similarity within the clusters (intra-group compactness) and at the same time having a high dissimilarity between the clusters (inter-cluster separability). The common criterion for grouping objects is their similarity, which is very often based on a predefined distance function depending on the data matter. Many algorithms exist and the choice of clustering algorithms depends on the type of input data and on the purpose of the clustering. In general, it is difficult to assess the effectiveness of unsupervised methods (including clustering) as it is a subjective matter [HTF09].

For multi-dimensional categoric (including binary) and continuous data, there are several functions that are used on a regular basis in clustering (e.g., Euclidean, Minkowski, TaniMoto, Mahalanobis). However, for complex data objects such as pictures, videos, graphs, trajectories, 3D Models, specialized distance functions need to be defined and used. In these cases, the objects are often described by so called feature vectors – multidimensional continuous/categoric vectors, which are used for measuring similarity between objects by applying the selected standard metrics mentioned above. For the thesis, the definition of graph and trajectory similarity are of special interest. The feature sets applied on these types of data objects are described in Sections 3.6 and 4.6 respectively. For defining *similarity between sets of objects (clusters)*, there are several standard approaches including single linkage, complete link, average link.

Clustering techniques can be broadly categorized into partitioning, hierarchic, density-based, grid-based, statistic model-based and neural network methods. The allocation of the techniques is not unique as various algorithms combine multiple methods for gaining better clustering results [HK06]. In the following we concentrate on neural network methods, in particular self organizing maps as they are relevant for the approaches applied in the visual analysis of both data types of interest to the thesis.

Self-organizing map (SOM) is a neural network learning algorithm with a strong disposition for visualization [Ves99, Koh01]. SOM combines dimension reduction and clustering. It preserves topological properties of the data set (i.e., two data points that are close in original space are close in the lower dimensional space). The

algorithm can handle large data sets and offers good clustering results. SOM method for clustering has previously successfully been applied to many different data types including documents [HKLK97], audio [RM06], and images [Bar08].

In this algorithm, a network of prototype vectors is iteratively trained to represent a set of input data vectors. The network is often assumed to be a 2-dimensional regular grid. During training, the algorithm iterates over the input data vectors. For each input vector, it finds the best matching prototype, and adjusts it as well as a number of its network neighbors toward the input vector. In the course of the process, the considered neighborhood size and the strength of the adjustment process (learning rate) are reduced. The training result is a set of prototype vectors representing the input data. In addition, the low-dimensional arrangement of prototypes on the network yields a topological ordering of the prototype vectors, approximating the topology of data samples in original data space.

The main parameterization required by the algorithm includes the initialization of the prototype vectors and the specification of the learning parameters. The latter includes the duration of the training process, the definition of the neighborhood kernel, and the degree of vector adjustment (the learning rate). While a number of rules of thumb exist for the parameter setting [Koh01, Ves99], finding good settings for a given data set usually requires experimentation and evaluation by the user.

In respect to the complexity of the computation, several enhancements of the original SOM calculation have been proposed [STM*06, Koi94, LAR99].

It is difficult to compare the effectiveness of self-organizing maps to other clustering methods. Preliminary comparisons to K-Means algorithm [Ult95] show that SOM yields better clustering results. As for limited data examples, they do not generalize to all data sets. However, self-organizing maps and k-means relate. When setting the neighborhood size to zero, SOM equals K-Means algorithm [Kas97].

Clustering quality The quality of the clustering results plays an important role when choosing the appropriate data representation. Various approaches to the algorithmic assessment of clustering quality are described in several surveys including [Pö4, LJB06, KL96, FFG*08]. From the portfolio of the proposed measures, a subset is applicable to multiple clustering algorithm results (e.g., quantization error, compactness, proximity) and the rest only to SOM results (e.g., topographic product, topographic error). The general measures focus on the assessment of cluster compactness and inter-cluster separation, while SOM specific measures also consider specific properties of the SOM output, in particular preservation of topology.

Visualization of Clustering Results Visualization is often key to understand otherwise possibly abstract clustering results. While certain clustering approaches implicitly yield visual representations (e.g., dendrograms for hierarchic clustering [SS02]), many other clustering techniques need a specific post-processing of the results in order to visually represent them. In case of high dimensional data outputs (e.g. multi-dimensional feature vectors in k-means clustering), parallel coordinate or star views [FWR99, Kan01] or projection-based approaches are common [EC01]. When focusing on SOM result visualization, the visualization techniques usually show the SOM grid with the reference prototypes (e.g., using multi-dimensional techniques), by labeling the cells with the most common members [Kas97], or by showing the nearest member to the prototype. An overview of the SOM visualization techniques is provided in [Ves99]. The visual assessment of clustering quality is supported by the display of the distances between SOM cells using the so called U- and U*-Matrix [Ult03], the exploration of the topology properties using vector fields [PDR06], or showing the SOM topology by nearest neighbor connections [PRD05]. The distribution of the prototype values across the SOM grid is presented using so called component planes [Koh01], where each dimension of the prototype vectors is shown in a separate heat-map.

For the exploration and refinement of the clustering results, interaction techniques such as focus, zoom, filtering, or multiple-linked views are provided [SS02, CL03, NHM*07].

2.5.3. Application-dependent Data Processing Techniques

A broad range of calculation/data processing techniques specific to their area of application exist, for example the calculation of integrated shares in the analysis of shareholding structures, simulation techniques in physics and technology, specific models in meteorology are also used in data processing and in Visual Analytics. They are usually based on application specific theories and algorithms. They can be used to support the analysis of the data in specific cases (see for example Section 3.7.3.1 on analysis of shareholder networks). The overview of these techniques in connection to Visual Analytics would go beyond the scope of this section. Please note, however, that there are few such techniques already used in the visual analysis of financial and economic data. WireVis [CGK*07] concentrates on the analysis of financial transactions for fraud analysis. Brath et. al. [BB03] present several visual analysis techniques for solving problems in the analysis of business data. Ziegler et al. [ZNK07b], [ZNK07a] apply specific time-series analysis methods for visual analysis of financial time series. The application of specific economic profit-maximizing decision strategies in Visual Analytics system for helping economic decision making has been shown in [SME08].

2.5.4. Use of Data Processing Techniques in Information Visualization

Recently, literature on information visualization has presented more intensively approaches, which include data processing (e.g., data or dimension reduction) as an initial step in the visualization. These approaches usually simplify the original data set before applying the data visualization techniques described in Section 2.3.3 (see Figure 2.11). In these cases, only the derived data is displayed.

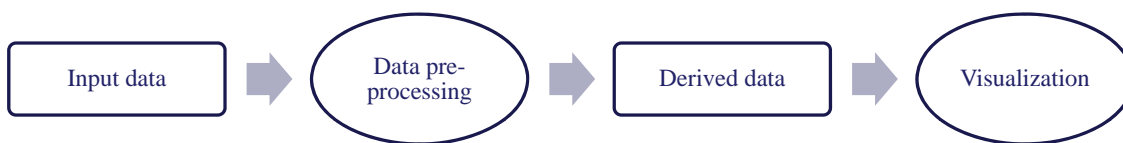


Figure 2.11.: Use of data processing in information visualization.

These approaches started to appear as an answer to data complexity and scalability issues which could not be solved by applying appropriate data visualization and interaction techniques. The main difference between such visualization techniques and visual data mining is that the data processing is performed automatically at the beginning of the visualization process (as a part of the visualization process) without user influence on the type of processing (and often its parameters).

Examples of such combinations of data processing and visualization techniques in the literature are:

- dimension reduction techniques (e.g., PCA, MDS [BCLC97], Sammon's mapping [HHW05], radial transformation of dimensions for showing multidimensional functions [Gof99]),
- data transformation from one data type to a more simple data type and/or less data items (e.g., transformation of time series into correlation matrices [May07] or correlation graphs [KG06], time series into growth rate matrices [ZNK07b], matrix of moving averages of time series [Pic95]),
- clustering (e.g., SOM [Š03], hierarchic aggregate [KSS06a], graph aggregation [KG06], hierarchic clustering and aggregation in parallel coordinates [FWR99]),
- sampling (e.g., [LCZ05], in graphs [LF06]), down-sampling of time series using identification of points of interest (e.g., [FCL*05]),

- reduction of number of edges or nodes in graphs using graph topology indicators (e.g., minimum spanning tree [MLM04]),
- mixed various methods (e.g., [CGK*07]).

3. Visual Analysis of Weighted Directed Graphs

3.1. Introduction

The analysis of large graphs plays an prominent role in many fields of research and has been applied in many areas, such as finance, biology, sociology, transportation, software engineering. The proper understanding of global and local graph structures is an essential aspect of any investigation. The graph analysis may include many different tasks. The tasks may concentrate on the examination of values of graph properties, on the occurrence of certain structures in the graph, on the effects of graph changes, on the comparison of many graphs and, depending on the use case, further analytical questions. The analysis can be performed at multiple levels of abstraction. We describe the tasks relevant to the thesis in Section 3.1.1.

The analysis of graphs is often supported by visual presentation of the graphs. Graph visualization research (see Section 3.2.3) mainly concentrates on the development of efficient graph layouts that allow fast and understandable presentation of the data, not explicitly on the support of particular analytical tasks. In large datasets, however pure visual inspection of the data may not be sufficient. Visualization technologies must therefore go beyond the simple illustration of the interdependencies and should be combined with further technologies for suitable visual graph analysis. Against this background, in this chapter, we present new ways of effective combination of graph processing and graph visualization techniques for supporting selected graph analysis tasks.

3.1.1. Tasks

In the visual analysis of graphs, we will concentrate on three types of tasks: examination of graph structure, analysis of subgraphs (motifs) in networks, and assessment of collections of graphs (see Figure 3.1). These tasks cover a broad variety of analytical questions in many applications and can thus be seen as exemplary.

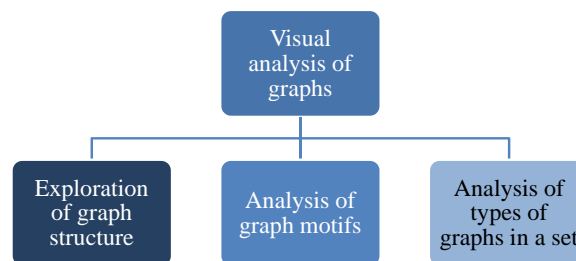


Figure 3.1.: Selected visual graph analysis tasks in the focus of the thesis.

3.1.1.1. Exploration of network structures

Within the broad variety of exploratory tasks [GFC04], we focus on tasks connected to the general assessment of the network structure and the basic characteristics of paths in graphs (e.g., the existence of a path between two nodes, the reachability of nodes from a give node etc.). For instance, in shareholding networks, it is interesting to know which companies are held by an entity (reachability), or whether two companies are connected and how (connectedness).

3.1.1.2. Analysis of graphs for the occurrence of certain substructures (motifs)

Network motifs are graph patterns, which capture important functional information in the network, or which occur with a higher frequency than it would be expected for random graphs. For example, in a shareholding network, motifs can show companies with many subsidiaries or structures leading to strengthening of the voting power in a company via holding shares in third companies. In gene-regulatory networks (see [Sch08]), the so-called “feed forward motif” has information filtering properties.

In particular, we focus on three motif-based network analysis scenarios:

1. Analysis of networks for occurrence of user-specified motifs in static graphs;
2. Analysis of network changes in dynamic graphs and their impact on identified motifs;
3. Analysis of network structures on multiple levels of detail based on (motif-driven) graph aggregation.

In the first case, it may be interesting to investigate which structures occur, how often they occur and where they occur in the network. For example, we can identify whether there are cross-holding structures or companies holding shares in many other companies. In phone call networks, we can detect persons who communicate with many people.

In the second case (data driven or user defined graph changes), the analysis focuses on the determination of significant changes in relationships. In particular, the analysis of the implications from data changes on graph structures is interesting. For example, in shareholding structures, buying stocks of one company may indirectly lead to gaining control over another company. In social networks, if an important person leaves the network, it may be crucial for the whole community.

Thirdly, in certain scenarios, the analysis of structures created between local relationships may be of interest. For instance, in order to investigate whether and what types of connections exist between specific functional structures. Such analysis can be of relevance at multiple abstraction levels.

3.1.1.3. Assessment of many graphs

In addition to looking for the occurrence of local substructures (motifs) in graphs, it can be necessary to analyze groups of graphs. In this case, it is often required to know what types of graphs occur very often or are atypical. For example, the analysis can involve intriguing questions such as: *Are there graph structures typical to the data set? Which types of structures are exceptional? How frequent are particular structures?*

The individual graphs analyzed can stem from many observations (e.g., networks in various geographic locations) or can result from partitioning (e.g., into weakly connected components) of one big graph. We apply the developed techniques on the latter case.

3.1.1.4. Combination of tasks within an analytical process

During the analysis process, the tasks entailed by the visual analysis of graphs can be (repeatedly) combined on demand. For example, the analysis may start with the exploration of the structure of one network by analyzing the connections of one entity. It can be followed by the analysis of types of substructures occurring in this network and then by analyzing the same issues in another graph (see Figure 3.2a for an illustration). Another analyst, however may decide to first analyze what types of graphs occur in the whole data set and after having found several interesting graphs, explore them individually in more detail (see Figure 3.2b for an illustration). Against this background, it is necessary to provide a flexible system supporting the combination of subtasks.

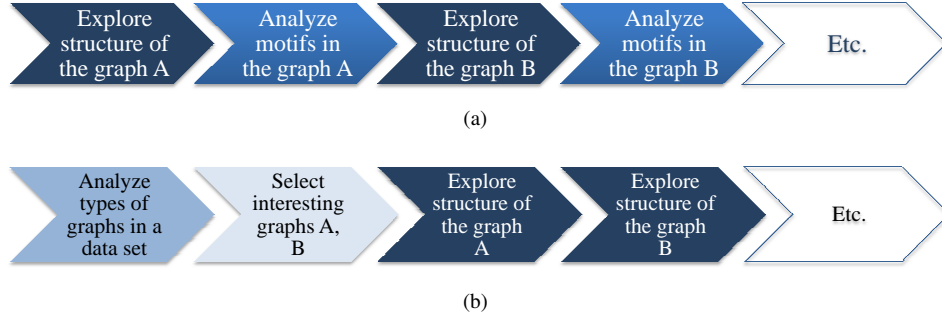


Figure 3.2.: Two examples of possible processes in the analysis of weighted directed graphs.

3.1.2. Contribution

This chapter shows how graph visualization can be enhanced with algorithmic graph analysis in order to support the various analytic tasks. In order to support exploration of graph connections, several graph algorithms for path and connectivity analysis are applied. Similarly, for analysis of graph substructures, motif search is employed. Visual motif-analysis is thereby extended with flexible motif definition, visual graph editing and graph aggregation. For the analysis of many graphs, we introduce a method for interactive visual clustering of graphs offering various possibilities of visual inspection of the data space.

3.1.3. Chapter overview

After presenting relevant work from the area of graph visualization, graph analysis and visual graph analysis (see Section 3.2) we present our approaches for the tackling of the above mentioned tasks (see Section 3.3). The detailed description of the techniques developed is provided in the Sections 3.4, 3.5 and 3.6. The application of our methods (see Section 3.7) focuses on analysis of shareholding networks.

The work presented in this chapter has been partially published in [TK08], [vLGS09], [vLGRS09] and [vLKS*10].

3.2. Background

This section presents the main techniques for visual analysis of graphs with focus on tasks presented in the introduction of the chapter (see Section 3.1.1). We start with graph algorithmic analysis including graph theory as a basis. We then proceed with presentation of techniques for interactive visualization of graphs. As a consolidation, we then overview visual analysis techniques combining visualization and algorithmic graph analysis.

3.2.1. Definitions

Graphs are a prominent data structure within Visual Analytics and related research fields. Often, graphs are applied for describing relationships between entities. A graph refers to a collection of vertices (nodes) and a collection of edges that connect pairs of vertices. A *graph* is a pair

$$G = (V, E); E \subseteq [V^2]; V \cap E = \emptyset, \text{ where}$$

elements of V are vertices and of E edges [Die05].

Graph size is determined by the cardinality of E and *graph order* by the cardinality of V . A graph $G_S = (V_S, E_S)$ is a *subgraph* of a graph $G = (V, E)$, when $V_S \subseteq V$ and $E_S \subseteq E \cap (V_S \times V_S)$. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *isomorphic*, if there is a bijective mapping f between their vertices $f : V_1 \rightarrow V_2$ and $\exists e_1 \in E_1, e_1 = (v_1^1, v_1^2) \Leftrightarrow \exists e_2 \in E_2, e_2 = (v_2^1, v_2^2) : v_2^1 = f(v_1^1), v_2^2 = f(v_1^2)$ [KSS06b].

Graphs can be *categorized* according to various criteria, for example, edge direction, edge weight, node labeling, existence of cycles in the graph, etc. [HMM00]. Figure 3.3 illustrates these types of graphs. A graph with edges that have a direction is called a *directed graph* or *digraph*. If the graph edges have associated numeric attributes (e.g., real numbers), the graph is called *weighted*. In graph theory literature, directed graphs with weighted edges are also called *networks*. In information visualization, the term *network* is often used in a broader sense also including graphs with cycles. Cycles are closed paths in the graph, i.e., sequences of nodes following the graph edges, where the first node equals the last node. If categoric attributes are associated with nodes, the term *labeled graph* is used. Additionally, graphs can have multiple edges between the same pair of vertices or only a maximum of one. Graphs with possibly multiple edges between two nodes are called *multigraphs*. We concentrate on weighted directed graphs with a maximum of one edge between two nodes (i.e., *networks*).

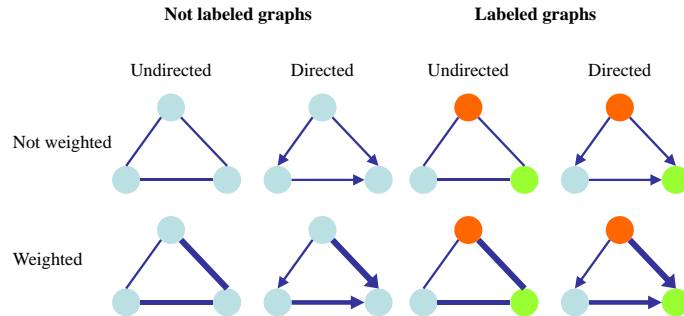


Figure 3.3.: Graph types. The edge weight is represented by edge thickness and edge direction by the arrows. Node labels are presented by node color.

The main classification in *directed and undirected graphs* is not sufficient if hierarchical and generic relationships exist within one graph at the same time. For example, in social networks, persons in an organization can be

in a subordination (hierarchic) relationship and at the same time in a friendship (generic) relationship. This type of graph in the following is referred to as a *compound graph*. Compound graphs can also be created by successive aggregation of graph vertices in a bottom-up approach. In this case, nodes (and implicitly, also edges) of the original graph are aggregated (i.e., merged), thereby creating constructed *meta-nodes* or *super-nodes*. The attributes of the meta-nodes are calculated from the attributes of the merged nodes. Similarly, edges between meta-nodes are aggregated into *meta-edges* and their attributes are calculated from the original edges. Compound graphs which are constructed in this way are also referred to as *aggregated graphs*. The type of calculation used is dependent on the particular application and graph type.

A *tree* is a graph without cycles. Trees are called *rooted* when one node is distinguished as a so called root node. Such trees are often treated as *hierarchies*, where the length of the path to the root denotes the level of nodes in the hierarchy. Connected graphs can be transformed to trees by removing edges in the cycles while the graph stays connected (i.e., there is an undirected path between all pairs of nodes) and includes all vertices of the original graph. This process can be reversed by adding back the removed edges. For weighted graphs (graphs with weight-attributes assigned to edges), algorithms for calculating minimum spanning trees (e.g., Kruskal's Algorithm [Kru56]) can be used for this task.

Graphs may also evolve over time, implying changes in the graph structure and/or in the attributes of vertices and nodes. If such a development is considered, we define *dynamic graphs* (i.e., time-dependent graphs) in contrast to *static graphs*. Time-dependent changes may affect the node/edge attributes, the graph structure, or both. If not stated otherwise, we concentrate on static graphs.

Figure 3.4 summarizes the graph classification presented above.

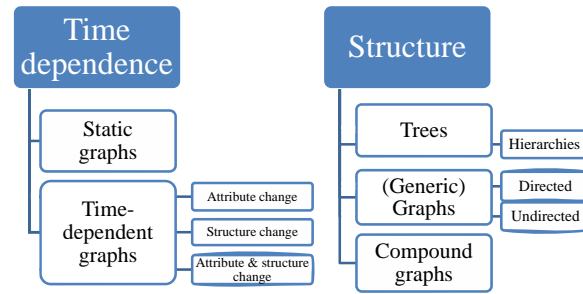


Figure 3.4.: Classification of graphs according to their time dependence and graph structure.

From the Information Visualization point of view, a specific group of graphs are *graphs with geographic reference*, such as transportation graphs. In this case, the nodes and possibly also edges of the graph have an inherent geographic location, which needs to be taken into consideration in their graphic presentation. For example, a specific graph layout algorithm is not needed for determining the position of each node on the screen. However, the fixed node position exacerbates graph readability problems, such as crossings and long edges. These problems need to be solved appropriately. Visualization of geographic data is a special research field, which we do not address.

Furthermore, graphs may be distinguished according to their *topological properties*. There exists a variety of literature on graph theory (e.g., [Die05]) which focuses on graph terminology, classification, and algorithmic graph analysis. In the following, we mention only the most relevant terminology. Basic graph properties include the number of nodes, graph density, and connectivity. Properties are often taken into account (or are a prerequisite) for certain visualization techniques. *The number of nodes* (i.e., graph order) often heavily influences which

methods can be used or fall short, with respect to readability and performance. Another important attribute is the *graph density*, the number of edges relative to the maximum potential number of edges. Sparse graphs have around $O(|V|) < |E| < O(|V|^2)$ edges, while dense graphs show density values close to one. Graphs with the maximum number of edges are called *complete graphs*. A *clique* is a subset of a graph that is fully connected. Large and/or dense graphs pose a scalability problem in visualization owing to limited display space and human perception capabilities. Several special graph structures appear often in real-world cases, and dedicated visualization methods have been developed for these [ACJM03, vHW08, JHGH08, MJW*09]. For example, in the so called *small world graphs* often found in social networks, most nodes are connected to each other with short paths. *Scale-free networks*, e.g., protein networks or certain types of social networks have degree distributions following approximately the power law.¹ *Bipartite graphs* are graphs whose nodes form two disjoint sets V_1 and V_2 , $V_1 \cup V_2 = V$, such that every edge $e = v_1, v_2 \in E$ connects vertex $v_1 \in V_1$ with one vertex $v_2 \in V_2$.

3.2.2. Algorithmic Graph Analysis

Many tasks connected to examination of large graphs can be supported by *algorithmic graph analysis* (see e.g., [Die05, Cal07, BGW03]). In this research area, many algorithmic solutions to graph analysis exist, for example for enumeration (counting graphs meeting specified conditions), finding a fixed graph as a subgraph in a given graph, routes and network flows (finding shortest paths or maximum flows between nodes), determination of network properties (e.g., diameter, maximum path, etc.), graph labeling (assigning labels to vertices meeting certain criteria, for example, graph coloring), graph clustering (either finding groups of similar nodes within one graph or grouping of several graphs depending on their similarity), graph matching (comparing similarity or exact match of graphs) etc. Please note the two meanings of graph clustering. In the following, we refer to the latter meaning – grouping of graphs.

In visual graph analysis, algorithmic graph analysis is often used for graph pre-processing. It includes graph simplification to reduce the order/size of the graph, while maintaining the main graph structure. For example, the reduced graph is used then for an easier visual inspection as large and complex graphs are difficult to understand even using advanced node and edge positioning algorithms (layouts). Such preprocessing steps can usually be performed automatically without user interaction. There are two main approaches to graph reduction: graph filtering [LF06, JHGH08] and graph aggregation [EDG*08]. In graph aggregation, nodes and edges are merged to single nodes and edges, thereby reducing the size of the graph and revealing relationships between groups of nodes. Graph aggregation can be repeated multiple times, creating a compound graph.

In the following, in correspondence with the defined tasks (see Section 3.1.1), we concentrate on three graph analysis areas: graph paths and connectivity, subgraphs and multiple graphs. For more in-depth examination into graph theory and analysis, we refer to the referenced literature.

3.2.2.1. Analysis of Graph Paths and Connectivity

When examining a (directed) graph, the interest may be put on examination of existence (reachability) and length of connections (paths) between nodes in the graph.

A *path* P between two nodes V_1 and V_n is defined as a sequence of nodes

$$P = \{V_1, V_2, \dots, V_n\},$$

such that there exists an (directed) edge between each two following vertices in the sequence. If $V_1 = V_n$, the path is closed (also can be called *cycle*).

¹Power law of graph degree distribution means that the fraction of nodes P_k that have k adjacent edges has an asymptotically power distribution $P_k \sim k^{-\lambda}$, where λ is a constant usually in the range $2 \leq \lambda \leq 3$.

In order to assess the *reachability* from one node (existence of a paths between the node and other nodes) breadth-first or depth-first search algorithms can be used [BG07]. Breadth-first search can be used also to find shortest path between two nodes (in unweighted graphs). In weighted graphs, specialized *shortest path* algorithms such as Dijkstra [Dij59], Bellman-Ford [For56, Bel58], Floyd-Warshall [Flo62] can be applied. Dijkstra algorithm is suitable for finding shortest path to all reachable nodes from one node, when edge weights are non-negative. The Bellman-Ford algorithm is also suitable for solving all shortest paths from a single node in a weighted directed graph. This algorithm allows for negative edge weights if a sum of weights a cycle is non-negative. This algorithm has longer computational times than Dijkstra algorithm, so it is mostly used only when negative edge weights are present. The Floyd algorithm is suitable for finding shortest paths between all pairs of nodes.

We can also assess the *connectedness* of a graph. A graph is connected if every node is reachable from every other node (in undirected graphs). For digraphs, we distinguish between strong and weak connectedness. A digraph is strongly connected if every vertex is reachable from every other vertex using the directions of the edges. A digraph is weakly connected if it is connected disregarding the edge directions. Weak connectedness can be examined by application of breadth-first search algorithms, strong connectedness by Tarjan's algorithm [Tar72].

3.2.2.2. Analysis of Subgraphs (Graph Motifs)

Motifs are predefined graph patterns. Usually, one is interested in those motifs that capture important functional information of a network or occur with higher frequency than it would be expected for random graphs. The space of possible motifs of a certain size (according to number of vertices) in directed graphs contains all possible combinations of edges for such number of vertices (see [Cal07] for example of motifs with size 3).

The algorithmic determination of motif frequencies for all possible motifs of a certain size is a NP-hard problem [GK07]. In general, exact search is preferred [SS05, WR06]. However, in many approaches, heuristics are used in order to accelerate the analysis [Sch08, Wer06]. These heuristics are usually developed for finding all possible motifs of a certain size. In our case, however finding of all motif occurrences is relevant and concentration on one selected (pre- or user-defined) motif is sufficient. Therefore, the exact search approach from [GK07] is mostly suitable for this type of task.

3.2.2.3. Analysis of Multiple Graphs

When analyzing multiple graphs, determination of *similarities between pairs of graphs* is a basis. The structural similarity can be determined both for labeled and unlabeled graphs. For labeled graphs, usually a matching of labels between two graphs is performed first and then structural differences are determined. There are two main approaches to the definition of graph structural similarity. Firstly, *transformation-based* approaches such as the Edit-2 Distance for undirected acyclic graphs [ZWS96]. This method calculates the distance between undirected acyclic graphs as the sum of costs when efficiently transforming one graph into the other. Secondly, *feature-based* approaches capture important data attributes in form of a feature vector or histogram (such as the graph histogram technique [PM99]). Feature vectors consist of values describing properties of the graph. In this case, distances between data elements are calculated using vector-space distance functions. The selection of features, in general, depends on the type of network (directed vs. undirected, weighted vs. unweighted, with vs. without node labels, with vs. without node weights etc.). A set of features used in this thesis and their description can be found in Section 3.6.2.

The results of graph similarity determination between pairs of graphs can be used for analysis of structural differences both for a single pair of graphs and for multiple graphs. We concentrate on the latter.

3.2.3. Graph Visualization

The visualization of graphs is one of the main research areas in information visualization. In this section, we first shortly introduce this research area.

An overview of techniques for graph visualization (including trees, graphs and compound graphs) is provided in Section 2.3.3. Moreover, several dedicated surveys on graph visualization have been published [HMM00, DPS02, vLKS*10].

Graph visualization techniques can be classified according to the visual metaphor used into node-link, matrix or combined representation. A comparison of node-link and matrix techniques is presented by Ghoniem et al. [GFC04]. According to the study, the advantage of node-link diagrams is their intuitiveness, compactness, and better suitability for path following tasks. They are more effective for smaller and sparse graphs. Matrix representations inherently do not have edge crossings and node overlapping problems, and are thereby suitable also for dense graphs. When using appropriate node ordering, they can easily reveal dense substructures in the graph. However, they also suffer from scalability in limited display spaces. In visual graph analysis, graph layout and matrix ordering influence the effectiveness of these representations. We concentrate on node-link representations in the following.

In node-link diagrams, the main challenge is the placement of the nodes so that graph readability and certain notions of graph aesthetics are supported. Typical requirements state that the nodes should not overlap, the number of edge crossings should be minimized, edge length should be homogeneous, and in general, that graph substructures should be easily recognizable. This problem is intensively studied in the *graph drawing community*. Given these aesthetic goals and constraints, the aim is to find algorithms that efficiently provide good solutions. An overview of graph drawing algorithms is given by Battista et al. [DBETT99].

The *graph layout* field is very large, and an extensive survey of proposed techniques is beyond the scope of this thesis. There has been a dedicated state-of-the-art report by Diaz [DPS02] summarizing techniques up to 2002. We can classify the techniques according to the type of node placement into force-based layouts, constraint-based layouts, multi-scale approaches, layered layouts, and further approaches. An overview of these approaches has been published in [vLKS*10]. Moreover, the related work part in [AAM07, MM08] as well as the comparison in [HJ07] nicely summarizes many currently available techniques. Force-based layouts rely on a simulation of mechanical laws by assigning forces among nodes and edges (e.g., [FR91, KK89, FLM95]). Constraint-based layouts extend the force-directed approach with constraints on node position (e.g., [DMS*08, DMW09b, DMW09a]). Multi-scale approaches first lay out a coarser graph (a subgraph of the original graph) and then include more nodes in a level-by-level fashion (e.g., [GK01, FT07, KCH02, HJ05, MM08]). Layered layouts (i.e., “hierarchic layouts”) place nodes of the graph on parallel horizontal layers (e.g., [Bab02, DK05, STT81]). Further approaches combine the previous techniques, or use completely alternative approaches to graph layouts such as projection of a node layout from high-dimensional to two-dimensional space [HK02], layout of the minimum spanning tree as a basis [ADWM04], or topologic properties of the graph parts, to choose the best graph layout [AAM07].

A specific field focuses on *visualization of multiple graph connected components*. In this area, first a layout for each individual connected component is calculated and then a specific placement of these components on the screen is performed. The mostly used placement method is called *packing*. It lays out the components so that they do not overlap and are space efficient. Dogrusoz [Dog02] compares several two-dimensional packing algorithms for graphs which use representation of graphs by their bounding rectangles. They include strip packing, tiling and alternate-bisection. The polyomino algorithm of Freivalds et al. [FDK02] uses polyomino representation of the graph objects, which substantially reduces the unused display space in comparison to rectangular shapes. Goehlsdorf et al. [GKS07] introduce new quality measures to evaluate a two-dimensional placement which yields more compact layouts than the previously mentioned approaches.

3.2.3.1. Interaction in Graph Visualization

An overview of interaction techniques in Information Visualization is presented in [KHG03]. Standard interaction techniques such as zooming, panning, brushing and linking [CMS99, War00] can also be applied in graph visualization. However, additional specialized interaction techniques have been developed for interactive visual graph navigation and exploration.

In line with the interaction taxonomy presented in Section 2.4.2, we categorize graph visualization interaction techniques according to whether the action of the user affects the data (the selection of the displayed data or changes to the data values) or the visual display of the data itself (visual parameters or visual representation). Please note that these two types of interaction are often closely connected. We mark such techniques with “(*)”. We briefly overview the techniques and refer to [vLKS*10] for deeper discussion.

Data selection techniques influence which parts of the data set are displayed. They may follow three graph exploration paths. Firstly, a top down approach starts from the whole graph and then constrains the part of the data set to be visualized by filtering according to criteria or by manual data selection. It offers an overview of the graph structure first and then concentrating on interesting parts. However, it may lead to occlusions owing to the limited screen size. Secondly, a bottom up approach starts from one selected node [vHP09, AF07] and successively shows more nodes/connections on demand. There are two main methods of choosing the additional nodes/edges to be displayed: based on graph structure, or based on a degree-of-interest function. At the beginning, only the most interesting part of the data set is visualized, however it is difficult to determine the starting point for the exploration and to define the degree-of-interest function. Thirdly, a middle-out approach combines both bottom-up and top-down approaches. It starts with a coarsened graph (middle) and then interactively either reduces or increases the graph coarsening level by hiding visible nodes or showing additional nodes [WMC*09]. The determination of the middle coarsening level and the next interactive steps poses the main challenge.

Changes of data values result from direct data value manipulation. Specifically, the user can change the data values on one level or create/change graph aggregations. In graph editing, the user can interactively delete or add nodes or edges directly in the visual interface. Graph editing affects the structural properties of the graph. Interactive graph aggregation is used for simplification of graphs. The graph aggregation can be predefined, or determined interactively by the user [AMA08, AMA09, HF06].

Changes of visual parameters affect the parameters of the visual presentation. They include highlighting of items, zooming, panning, view distortion, and other techniques. For graphs, specific techniques have been proposed. For example, guided panning allows to navigate along edges of a selected node and thereby to explore the structure of the graph [MCH*09]. Semantic Zooming(*) combines zooming with an increasing level of detail. In particular, graph aggregation can be used for gaining a coarser view on a large graph [EDG*08, AvH04]. Distortion techniques allocate more space to items in focused areas and thereby, improve the readability of the data of interest. They are used both for node-link and space filling graph visualization techniques.

Changes of visual scheme includes layout change and change of visual representation. Layout change, in node-link diagrams, affects the positions of the data items on the screen. It can be performed by changing of the layout type with automatic recalculation of the new layout, by manual movement of nodes, or by adjusting the layout parameters including automatic readjustment of the layout. When concentrating on user-defined changes to graph layouts, an approach to easy selection and layout change of nodes and subgraphs was presented in [MJ09]. Furthermore, interactive adjustment of the layout constraints was presented in [DMW09a]. Change of visual representation, e.g., from a matrix to a node-link diagram was presented in [ZMC05, HFM07]. This change can affect the whole data view [HFM07] or only a part of it [ZMC05, HFM07]. In order to be able to follow the changes, smooth animations across transitions should be used.

3.2.4. Visual Graph Analysis

Algorithmic graph analysis is beneficial during all stages of the visual graph analysis process. Relevant techniques allow, e.g., to reduce a large graph to a smaller graph prior to visualization, to search for specific graph structures of interest, or to find similarities and dissimilarities for generating comparative graph views. In this section, we describe important graph analytical approaches.

In this section, we present relevant works combining visualization and algorithmic graph analysis suitable for the three types of tasks addressed in this chapter.

3.2.4.1. Analysis of Graph Structure

In most user tasks, the analysis of the relationships between entities in the graph and the assessment of the global graph structure plays the key role. These tasks may be effectively supported by a combination of algorithmic graph analysis and interactive visualization. The algorithmic methods allow, e.g., to calculate node/edge properties, identify clusters in the graphs, etc., which results are visualized interactively. In the following, we summarize the methods according to user tasks starting from more simple to more complex tasks.

Identification of important nodes: In networks, some nodes play a specific role owing to their position within the network. For example, so called hubs and authorities can be identified and visualized in the network, enabling faster analysis of the graph [OPPROG09]. The importance of nodes and edges is measured by derived quantities such as centrality-based measures [Fre79] and ranking-measures [WS03].

Analysis of connections between two nodes: Besides focusing on single nodes, relations between two nodes can be analyzed, typically by calculation and highlighting of shortest paths between the entities. Usually, such analysis is combined with interactive selection of two entities of interest [TK08, HB05, HF07b].

Analysis of graph structure on several aggregation levels: User-defined or data-driven graph aggregation can reveal relationships between groups of entities in a graph. The grouping may be based on categoric node attributes [Wat06], or on a predefined node hierarchy [AMA09]. It can also be user-specified [AMA08].

Identification of the impact of graph changes on the structural properties: In time-dependent graphs, the role of the nodes can change over time, therefore analysis and visualization of topologic properties (e.g., betweenness centrality) of selected nodes has been proposed [PD08]. Additionally, when analyzing user-defined changes (in what-if-scenarios) the impact of node or edge deletion/addition on local substructure can be analyzed and highlighted.

3.2.4.2. Motif-based Visual Graph Analysis

The *analysis of a graph for motifs* is applied mainly in biology and chemistry [Sch08], as motifs often play an important role in biologic reactions. There are several tools for analysis of graphs for motifs from this area. Some of them also offer simple visualization of the motif results. The MAVisto tool [SS05] offers motif search function with the display of motif frequency also in comparison to randomized networks. The distribution of a particular motif can be shown using a specified layout. The FANMOD application [WR06] allows for fast detection of networks motifs and display of found motif types with their frequencies. The functionality of the system is similar to Mfinder, mDraw, and SNAVI ([MFi, MJW*09]). These approaches however are computationally intensive (search for all possible motifs) and either not offering visualization of the graph at all (only the list of graphs) or restrict the drawing only to small graphs (up to ca 100 nodes). Similar to motif-based analysis, the power graph analysis [RRAS08] examines the network for selected specific node groups (stars, cliques and bicliques) and uses them for graph simplification.

When looking at the *use of graph motifs for graph visualization*, a graph layout based on subgraphs was introduced by Holleis et al. 2005 [HZG05]. Their approach focuses on subgraph layout first and then introduces connections between the subgraphs. It was also used for visualization of motifs. Additionally, a motif-preserving layout based on force-direction was presented in [KSS06b]. The authors propose to analyze a network for occurrence of various motifs and to show all types of found motifs in a specific view. They offer the possibility to show matches of selected motifs in the main network view. Huang et al. [HMS*05] present visualizations that highlight motifs found in a network. They concentrate on non-motif parts of the graph by firstly simplifying the graph motifs and then showing only the simplified motifs. While they concentrate on a small set of predefined motifs, they present specific simplification algorithm for each motif type. The main constrain of the system is the concentration only on predefined motifs. The motif visualization shows the simplified motifs on one graph layer and the non-motif edges and vertices on a second layer. This creates a 2.5D view on the graph. For the separation of various motif types separate planes in 2.5D and color coding is used which, as claimed by the authors, reduces the comprehensibility of the graph. Alternative views provide motif disconnection using node duplication and placement of motifs inside colored spheres. This approach may contribute to better readability of the motifs, however the use of node duplication and the need for their connection by new edges leads to more complex graphs to be visualized.

3.2.4.3. Visual Analysis of Many Graphs

One specifically important analytical task is the examination of the similarities and differences between multiple graphs, especially focusing on structural aspects. Usually, structural differences are in the focus. Such difference may be identified by the identical node labels in both graphs, or by graph matching algorithms. After the matching, visualization is employed to explore the differences [AWW09]. There are various types of analysis which we describe next.

One-to-one node comparison of two graphs Probably the most common task in graph comparison is the matching of individual nodes from one graph to individual nodes of the second graph. The VisLink visualization approach [CC07] was developed to support this task. It shows both graphs on separate planes in 3D, and draws matching links between corresponding nodes. For comparison of hierarchies, a similar approach, based on drawing the two hierarchies in opposite parts of the display and linking of their leaf nodes was proposed in [HvW08]. In both cases, the visibility of matching links can be increased by edge bundling.

One-to-many nodes comparison of two graphs: One-to-many nodes comparison concerns correspondence of one node in one graph to many nodes in another graph. Di Giacomo et al. [GDLP09] developed a system that visualizes these one-to-many connections with low overlapping of links.

Structural differences between two graphs: When analyzing structural differences between two graphs, analysts are often interested in identifying which links or parts of the graphs correspond to or differ from the other one. For the analysis of trees, the TreeJuxtaposer system supports to analyze and highlight structural differences between two trees [MGT*03]. For general graphs, Fung et al. [FHK*09] use both multi-level graph views following the VisLink approach [CC07], and overlapping of two networks with highlighting of common structural parts. Archambault [Arc09] uses graph aggregation and graph filtering to reveal structural differences between two graphs.

Comparison of multiple graphs: Visual Analysis of many graphs is by now restricted to either algorithmic analysis or visualization as presented in Section 3.2.2. The interactive combination of both is relatively rare, though some of the above mentioned approaches use simple visualizations for presenting results of algorithmic analysis or the visualizations use selected algorithmic methods. Self-organizing map (SOM) clustering for graph matching has been used by Gunter and Bunke [GB02]. They use edit-based graph distance for the recognition of

handwriting. However, this approach has not been applied to general graphs and moreover, their work does not include visualization of clustering results. Neuhaus and Bunke [NB05] also use SOM with graph edit cost for graph matching applying the approach to numerically labeled graphs.

3.2.5. Summary

Graph visualization concentrates on effective presentation of graph data, in particular on graph layouts and graph interaction. The visualization techniques can be used for visual graph analysis when combined with further suitable algorithmic analysis techniques.

Analysis of Graph Structure Visual analysis of graph structure includes inter alia identification of important nodes, analysis of connections between two nodes, analysis of graph structure on several aggregation levels and identification of graph changes. Important nodes are usually identified based on their topologic properties. The analysis of connections has concentrated only on calculation and highlighting of shortest paths between nodes, graph aggregation follows mainly either a predefined hierarchy or a user-defined graph hierarchy. Graph changes analysis mainly concentrates on the changes of topologic properties of single nodes.

Motif-based Visual Analysis There are several methods for analysis and visualization of graph motifs and general graph visualization techniques. Graph drawing approaches often disregard graph substructures and do not offer analysis possibilities. Motif-based approaches are restricted to predefined motifs or search for all possible motifs at the beginning of the analysis (being computationally demanding).

Visual Analysis of Many Graphs The analysis of graph similarities has mainly concentrated on examination of structural differences between two graphs. The identified pairwise differences/commonalities are then displayed in a side-by-side view. In case of a large sets of graphs, visual inspection of the sets of graphs or algorithmic analysis of similarities is employed. The visualization uses space efficient placement of graph components on the screen. The current placement methods however disregard similarities between graphs and suffer from scalability issues for large sets of graphs. The algorithmic analysis misses tight coupling with visualization.

3.3. New Approaches to Visual Analysis of Weighted Directed Graphs

In order to address the three analytical tasks in visual analysis of weighted directed graphs (presented in the introduction of the chapter), we present three integrated approaches which enhance visual presentation of the data with graph algorithmic analysis, and, if appropriate, also tracking of the analytic process (see Figure 3.5).

For the exploration of the graphs, we combine state-of-the-art interactive visualization techniques with presentation of the results of algorithmic graph analysis (in particular path and connectivity algorithms). This provides a basis for the other two approaches. The visual analysis of graph motifs also combines interactive visualization with motif search in static and dynamic case. Moreover it provides (motif-based) graph aggregation for graph simplification and analysis of graphs on multiple abstraction levels. The analysis of many graphs is supported by feature-based clustering of graphs combined with interactive visualization of clustering results together with visual assessment of clustering quality. Owing to the complexity of the tasks and flexibility of their combination, the latter two approaches also offer tracking of the analytical process for the reproducibility of the results.

We explain these approaches in more detail in the next subsections.

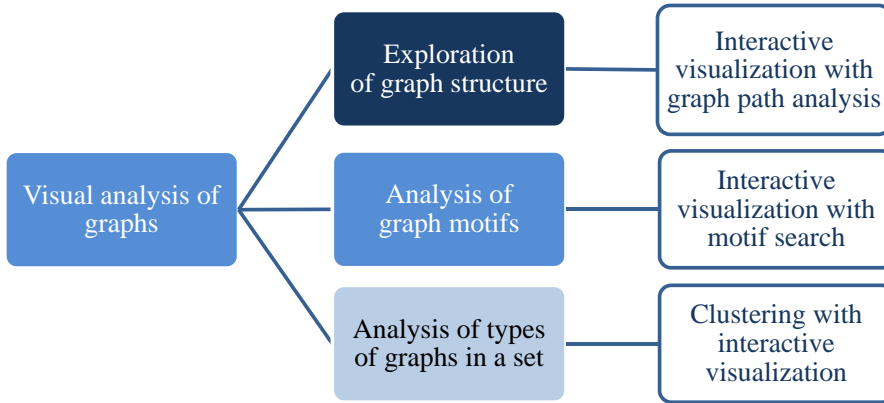


Figure 3.5.: The proposed approaches to the visual analysis of graphs supporting three selected analytical tasks.

3.3.1. Approach to Interactive Visual Exploration of Weighted Directed Graphs

In this section, we propose an effective visual exploratory tool that supports the analyst in understanding the interdependencies in networks. Visual graph exploration usually relies on interactive graph visualization using efficient graph layout. The focus of our work is, however, not on development of new visualization techniques solely (including new graph layout algorithms), however on the support of analytic tasks. Therefore we use state-of-the-art graph interactive visualization technologies enhanced with algorithmic graph analysis methods for the support of exploratory analysis tasks (see Figure 3.6).

Interactive visualization of graphs employs node-link diagrams, matrices and combined methods as explained in the theoretic background (see Subsection 2.3.3.1) combined with visual interaction functionalities (such as zooming, panning, highlighting, move of nodes, visual mapping). In our approach, we use node-link graph representation of graphs with flexible (user-chosen) graph layout. The motivation for use of node-link diagrams is a broad familiarity of this type of graph representation to the users. Our approach is layout independent

and therefore flexible for application on various use cases. Our tool includes standard visualization interaction techniques for visual exploration as presented in the interaction taxonomy (see Figure 2.10). In addition to interactive visualization of the graph, our approach provides a variety of analytical exploratory functions with visual output. It offers, for instance, calculation and visualization of paths between nodes, and reachability of nodes from a selected node.

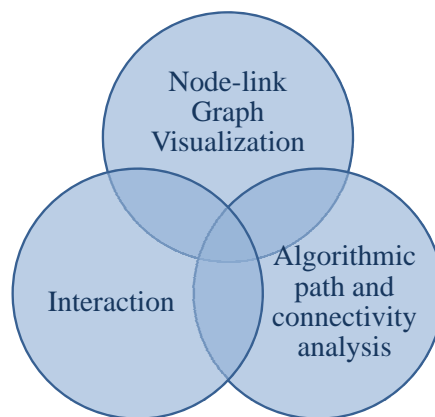


Figure 3.6.: An approach to visual graph exploration integrating interactive graph visualization and algorithmic graph analysis.

3.3.2. Approach to Visual Analysis of Graphs Using Motifs

Our approach to visual analysis of graphs using motifs is threefold (see Figure 3.7). These three sub-approaches are integrated into one system and can be used in combination on demand. Each sub-approach corresponds to one analytical task presented in Section 3.1.1. As a basis, search and visualization of found motifs on demand is provided (see Subsection 3.3.2.1). The search and visualization methods are applied also to visual analysis of graph changes (see Subsection 3.3.2.2). Finally, the visual analysis of large graphs is supported by motif-based graph aggregation (see Subsection 3.3.2.3). These approaches are detailed upon in the following.

3.3.2.1. Visual Exploration of Graph Motifs

Visual exploration of graph motifs usually uses graph visualization with highlighting of motifs found in the network (see Figure 3.8 for an illustration). These approaches however rely on a predefined set of motifs and use global motif search. In practice, the choice of interesting motifs is application, data and task dependent and may even change throughout an analytic process. Against this background, it is difficult to define a complete set of relevant network motifs in advance. Therefore, we also support user-defined motifs in addition to a set of basic graph motifs (see Figure 3.20 on page 71). In order to support an easy input of user motifs, these new motifs can be visually interactively defined by drawing. In addition, we employ local search for motifs (motifs including a specific node or edge), which allow focusing on specific parts of the network and at the same time require less computational time than searching in the whole network. In some cases, specific user-defined constraints on the motifs can be posed, such as minimum edge weight in the motif edges. Therefore we offer also filtering of motifs

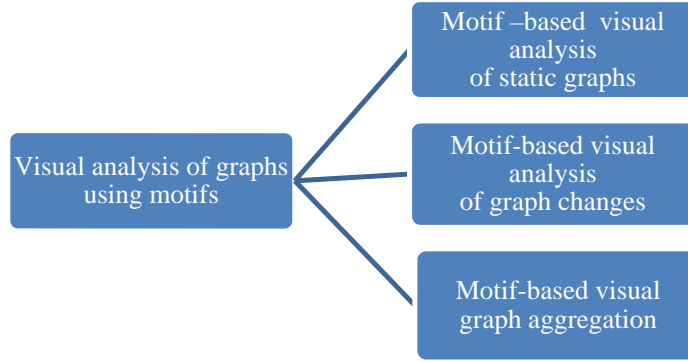


Figure 3.7.: Three types of motif-based visual graph analysis.

according to edge weight criteria. The comparison of our approach with state-of-the-art approaches for visual motif analysis is provided in Figure 3.9.

3.3.2.2. Visual Analysis of Graph Changes Using Motifs

Usually, when showing graph dynamics, changes in the graph compared to the previous graph are highlighted (e.g., new/changed edges or nodes). This might be insufficient from an analytical point of view, as for the analyst also the impact of those changes on graph structures can be of high interest. Changes in graph structures can be identified by analyzing the graph for appearance of new or deletion of existing local substructures (motifs). For this purpose, we combine motif analysis with interactive visualization thereby helping to discover indirect effects of graph changes.

The changes can be induced by source data or can be done on demand by the user (in the so-called “what-if-analysis”). In our approach we concentrate on the second case. This type of graph changes is more difficult as it requires also identification of implied graph changes (e.g., node deletion implies changes to the adjacent edges) and the need for tracking of analytic steps for reproducibility of the results.

In our work, we therefore enhance simple visualization of graph changes with identification of implied graph changes, tracking of analytic steps and visualization of impact of changes on local structures. This approach extends significantly other state-of-the-art approaches. The comparison of the two methods is provided in Figure 3.10.

3.3.2.3. Approach to Visual Analysis of Graphs using Aggregation Based on Motifs

Graph aggregation is often employed for visualization of large graphs as it provides simplification of the original graph. It thereby improves occlusion problems and offers more clear presentation of the data while keeping the structure of the graph. Suitable graph aggregation can support analysis on higher levels of abstraction. For example, in shareholder networks, groups of companies from each sector can be merged into single nodes thereby not only reducing the number of entities in the graph but also allowing for analysis of inter-sectoral relationships in the economy.

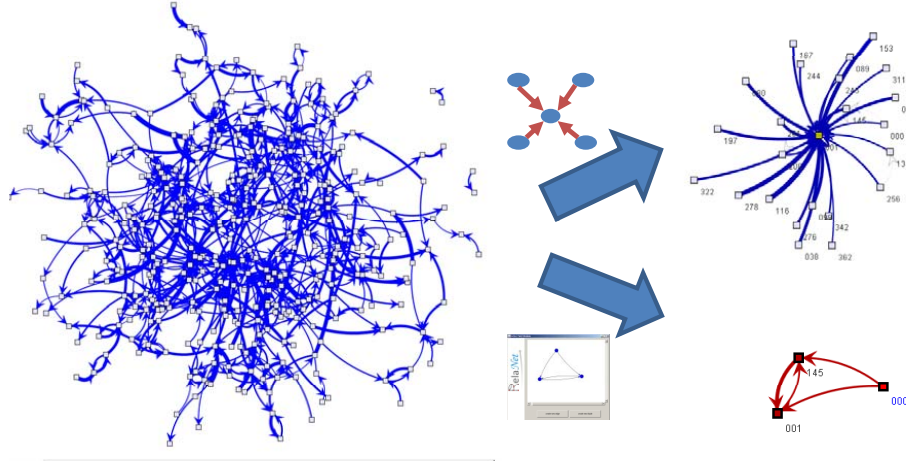


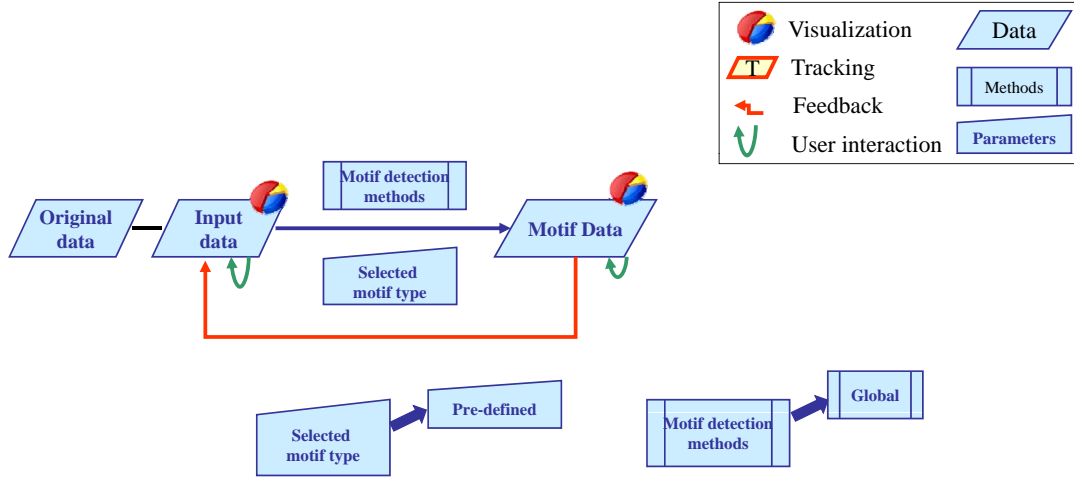
Figure 3.8.: An example of the use of visual exploration of graph motifs. Left: State-of-the-art visualization of the whole graph using a node-link diagram showing an overcrowded display, in which the entity relationships are difficult to explore. Right: The result of the new approach showing selected motifs found and filtered in the original graph. This view reveals possibly interesting substructures of the graph in a more easily interpretable way.

There are various methods for the choice of nodes to aggregate, e.g., according to the node properties (such as node betweenness), attribute properties (such as node attribute values) or a predefined node hierarchy. In addition, also interrelationships between local substructures in the network are important to analyze. Therefore, we present a new approach for the visual analysis of large graphs using hierarchic motif-based graph aggregation (see Figure 3.11 for an illustration). The main advantage of this method is the subsequent merging of local (functional) graph substructures of the input graph thereby revealing their relationships on multiple levels. Our approach is thereby not restricted to a set of predefined aggregation structures. In addition, we also track graph features (e.g., graph size, graph order, number of motifs) in each aggregation step for providing information on the structural changes in the network implied by the aggregation.

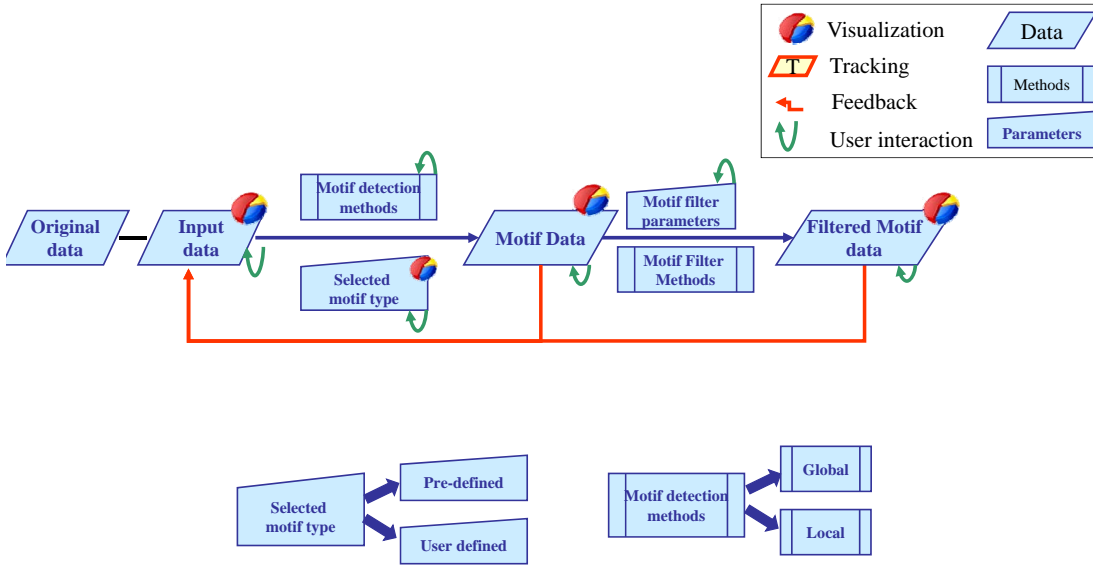
The aggregation *can be successively repeated* on the already aggregated graph. The choices of sequence of graph aggregation parameters (e.g., type of graph motif or node attribute) used for aggregation plays a significant role in the structure of the output graph. Therefore our approach includes tracking of the sequence of the aggregation steps (including their parameters). This should allow for reproducibility and better understanding of the results.

In this thesis, we employ the following three *ways of defining group of nodes for aggregation*, while concentrating on motif-based aggregation:

1. *interactively user-defined*: offers the possibility to flexibly analyze the underlying network without specific criteria. For example, the analysts may want to group companies according to results from previous analysis or her experience.
2. *based on node attribute values*: offers the possibility to analyze relationships between groups of nodes with the same properties. For example, the analysis of inter-sectoral or inter-regional relationships can be supported in this way.

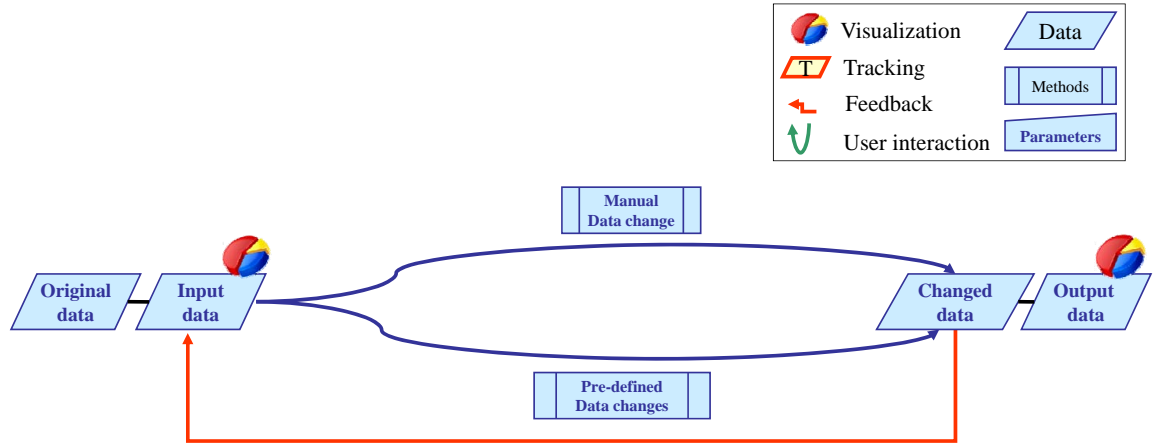


(a) State-of-the-art visual motif analysis using search for and highlighting of predefined motifs in the whole graph (global search)

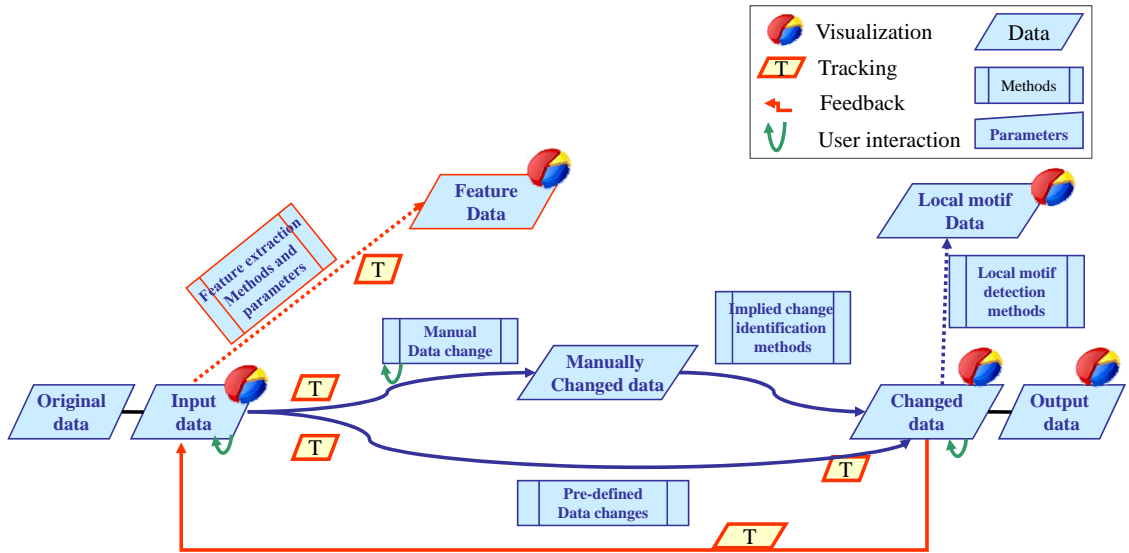


(b) New approach to visual motif analysis including also user-defined and local motif search and visualization.

Figure 3.9.: Approaches to visual graph motif analysis. Top: The state-of-the-art approach, bottom: the new approach. The new approach includes also search for user-defined motifs and local search for motifs.



(a) State-of-the-art visual graph change analysis showing only the manual or data-driven data changes without their implications.



(b) New approach to visual graph change analysis including also identification of implied graph changes in case of manual graph change, local search and visualization of changes in the motif graph structure as well as calculation and display of changes of graph features.

Figure 3.10.: Approaches to visual graph change analysis. Top: state-of-the-art approach, bottom: new approach. The new approach also identifies implied changes on the graph substructures and for manual data changes also the need for graph adjustment. It includes also tracking of changes of graph features owing to the data changes. In this way, the implications of the graph changes on the graph structure can be identified.

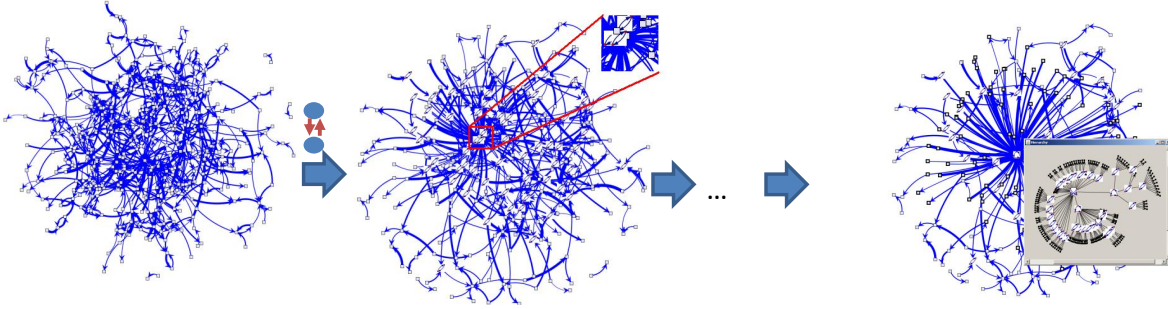


Figure 3.11.: An example of motif-based graph aggregation used for graph simplification and visual analysis of relationships between graph substructures on multiple levels of abstraction.

3. *motif-based*: is suitable for analysis of relationships between specific graph substructures. For example, it can be interesting to examine whether and what type of relationship there is between companies with many shareholders.

Figure 3.12 shows the currently used aggregation process (top) and our approach (bottom). Our approach includes also motif-based aggregation and tracking of aggregation parameters. Moreover, tracking of properties of the aggregated graphs at each abstraction level allows for analysis of structural changes owing to the aggregation. The results of each aggregation step are visualized using enhanced visualization tools based on the approach proposed in section 3.3.1 allowing for exploration of aggregation results.

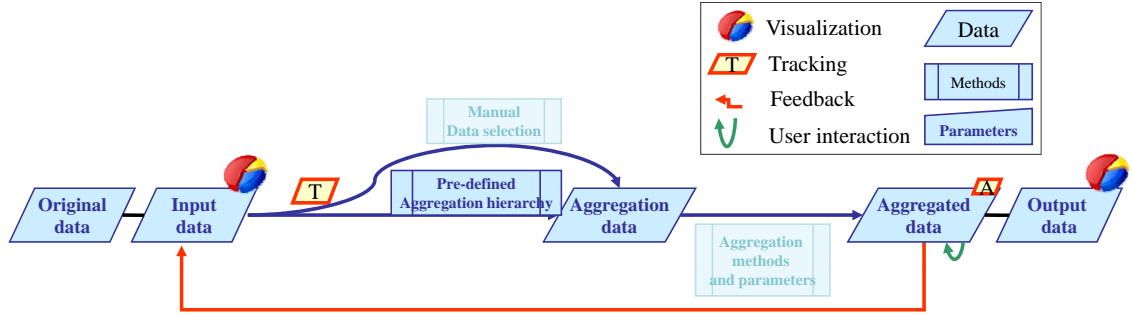
3.3.3. Approach to Visual Analysis of Many Graphs Using SOM Clustering

We now present the part of the system suitable for visual analysis of many graphs, in particular multiple connected components of one graph.

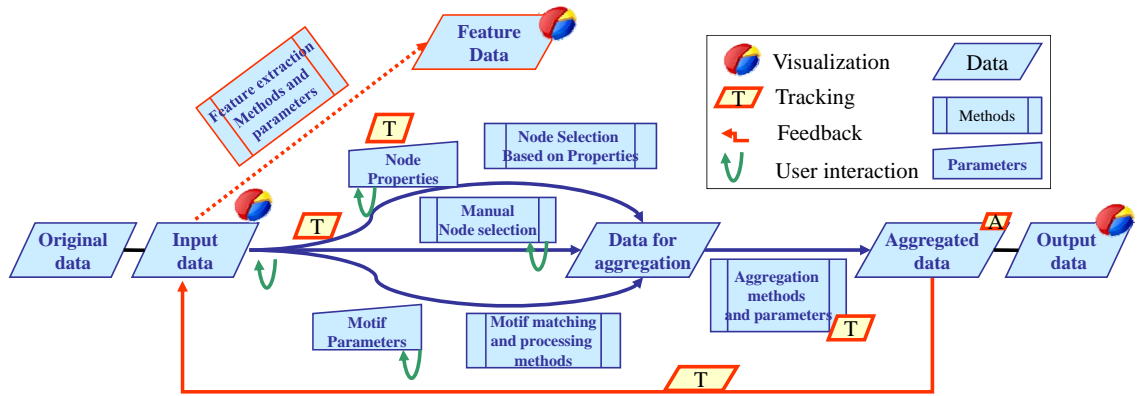
The visualization of large graphs (with tens of thousands of nodes or more) has recently received much research attention [AAM07, CZQ*08, GK01], and results of it have been applied to visual analysis of large network data sets from different domains. However, only few specialized approaches are available for visualization and analysis of graphs with multiple components [FDK02, GKS07]. These approaches do not focus on clustering similar components in order to assess the structure of the graph (see Section 3.2.3 for more details).

In our approach, we follow Keim’s Visual Analytics mantra “Analyse First – Show the Important – Zoom, Filter and Analyze Further – Details on Demand” [KKMT06]. In particular, we use clustering of graphs based on feature description of each graph in the data set for showing overview of the data set and subsequently allowing for deeper visual exploration of the data set (see Figure 3.13 for an illustration). The steps of the iterative clustering process are similar to the general clustering framework VISTA [CL03]. They include data pre-processing, initialization, clustering, quality assessment and post processing.

Clustering is an important data analysis technique. It supports the examination of large amounts of data by abstraction to a limited number of data prototypes describing groups of data and providing overview of the whole dataset. In particular, we employ self-organizing maps as clustering algorithm (see Subsection 2.5.2.1 for more details) because it is suitable for large datasets and provides reasonable results with direct visual output. The SOM clustering algorithm relies on an implementation of a *similarity* function defined over the set of data



(a) State-of-the-art visual graph aggregation with limited possibilities for selection of aggregation methods and data. It does not include hierarchic motif-based graph aggregation and tracking of aggregation process.



(b) New approach to visual graph aggregation including interactive definition of aggregation methods and data, hierarchic motif-based aggregation and tracking of the aggregation process.

Figure 3.12.: Approaches to visual graph aggregation. Top: state-of-the-art approach, bottom: new approach. The new approach enhances aggregation methods with hierarchic motif-based graph aggregation and tracking of graph feature changes resulting from graph aggregation. It allows to visually analyze relationships between substructures on multiple levels of abstraction and analysis of graph changes based on aggregation.

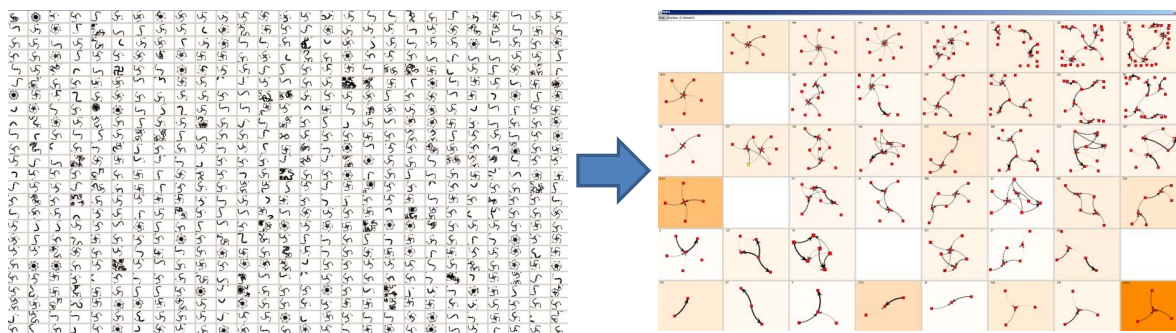


Figure 3.13.: An example of clustering of multiple graph components revealing types of graphs in a set. Left: A small part of the set of graph components to be analyzed using state-of-the-art graph visualization techniques. This view does not support assessment of types of graphs and their similarities. Right: Visualization of the result of clustering of multiple graph components using our approach showing types of graphs and their frequency (represented by background color) in the analyzed set of connected components.

elements which are to be clustered. There are various ways of defining graph similarity. In our approach, we use the feature-based technique.

Our clustering-based visual graph analysis (see Figure 3.14) starts with partitioning the whole input graph into its weakly connected components (in case the set of graphs is not already provided as input). Each extracted component (resp. graph) is described by a set of features creating a feature vector data set (see Section 3.6.2). The summary table of the feature data set or the whole feature vector data set can be explored using various views (e.g., using multivariate visualization techniques such as parallel coordinates). The following feature selection step is supported by a user interface for interactive weight adjustment (see Section 3.6.3). The feature selection and weighting performed influences the similarity function used in the subsequent clustering and thereby also the clustering results. The selected feature set is used as input for calculating similarity between components during clustering. After choosing the SOM clustering parameters, SOM clustering is performed. The clustering results are shown using interactive visualization techniques described in Section 3.6.5. The subsequent assessment of the clustering quality is supported by interactive exploration of the results as well as by calculation of the SOM quality measures and their detailed display (see Section 3.6.6). During the whole process, the system supports editing of user annotations. These annotations can be used to comment on decisions on process parameters or on intermediate analysis results obtained while working with the system. The results of the process (input and output data, annotations and parameters) are stored, supporting reproducibility and comparison of results. The feedback loop allows to change the parameters, switch between process stages and views and thereby create new results and insights.

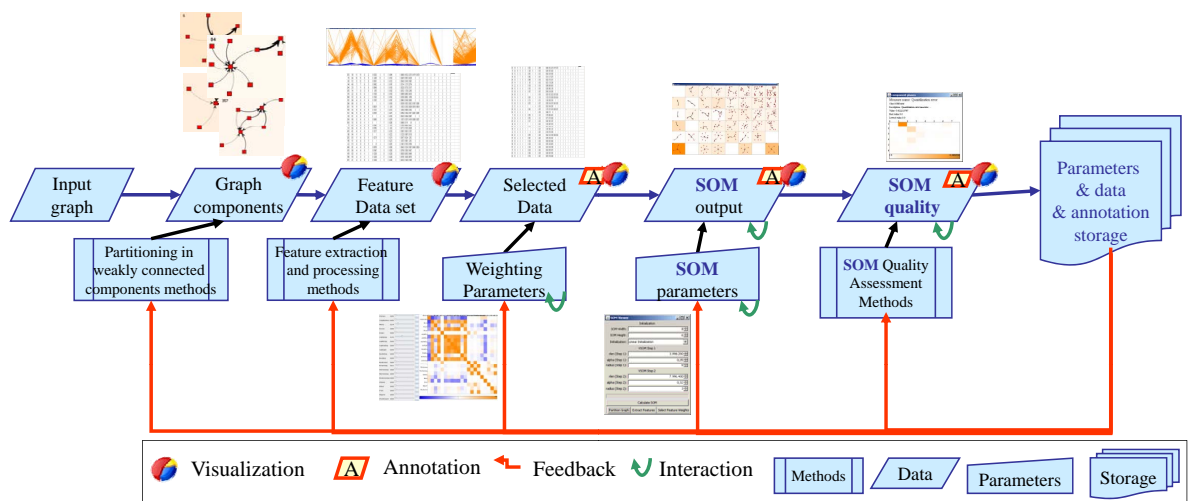


Figure 3.14.: The new system for visual analysis of multiple graphs. The system includes graph partitioning, graph feature extraction, SOM clustering and interactive visualization and assessment of results with feedback loop.

3.4. Interactive Visual Exploration of Weighted Directed Graphs

3.4.1. Introduction

In this section, we describe an effective Visual Analytics tool for visual exploration of the interdependencies in a network. In our work, we employ state-of-the-art graph visualization technologies (in particular node-link diagrams with standard graph layouts) which we extend with highlighting of results of algorithmic graph analysis methods. We first provide details on the interactive visualization system and then on the analytic exploratory functions included.

3.4.2. Interactive Visualization

Visualization In our work we employ node-link diagrams and use the JUNG [OFW, OFS] graph visualization library as a basis of our system. In order to support the analysis of networks of varying sizes and structures we do not restrict our system to one layout algorithm but offer a variety of algorithms for the user to choose from specifically for her needs (see Figure 3.15). The layout can be interactively adjusted by the user (using panning, zooming, dragging of the nodes, rotating, etc.).

Our system offers flexible assignment of input data to visualization parameters in order to be able to support analysis of various data sets and user foci. For example, the company name, the sector, the number of employees, total assets, etc.) can be mapped to node and edge attributes (see Figure 3.16). The input data is stored in a MySQL database and loaded on demand.

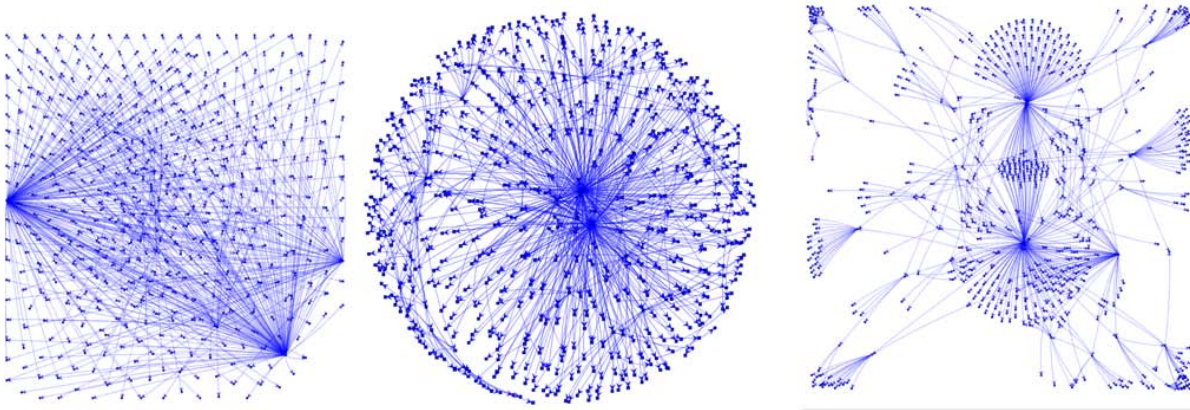


Figure 3.15.: Various layouts of the same graph. Left: ISOM layout [Mey98], Center: Kamada-Kawai layout [KK89], Right: Fruchterman-Reingold layout [FR91].

Interaction We integrated various visualization interaction functions supporting both data changes and changes of visualization (scheme and parameters). They include layout change, changing of visual mapping, zooming, panning, highlighting, filtering, and graph editing. They offer a possibility to interactively explore the data space and are essential part of the visual analysis of graphs. These visualization interaction functions are extended with analytical interaction functions described in the following.

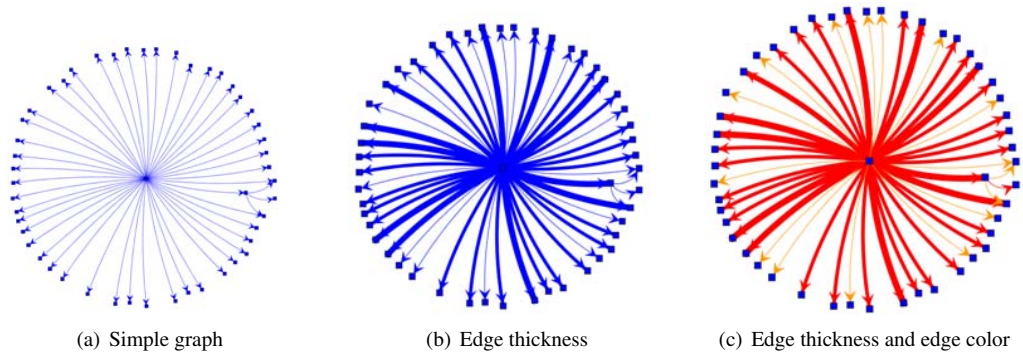


Figure 3.16.: Examples of interactive data mapping in the graph visualization supporting various analytical foci.

3.4.3. Visual Analytical Functions

In our work, we concentrate on visual analysis based on algorithmic graph calculation functions for exploration of paths and connectivity in the graph. The results of these algorithms are visualized using showing (display of the previously hidden items), highlighting (highlighting of the already visible items) or filtering (hiding of the items which are not in focus).

- For *calculation and visualization of direct and indirect children (parents) of a node* we use breadth-first and depth-first search algorithms. The successors (children) or predecessors (parents) are highlighted in the original graph (see Figure 3.17). In weighted graphs, constraints on edge weight (e.g., minimum edge weight) can be posed (see Figure 3.18).
- “*Out (in) flows*” are calculated depending on the application used. In general, common flow and length of path calculation algorithms (e.g., Dijkstra, Floyd, etc.) can be used. In our system we concentrate on shareholding networks. Therefore we employ specific algorithms based on the economic literature (see Subsection 3.7.3.1 for more details). In addition, we offer identification of flows obeying user-defined criteria on minimum edge weight useful in identification of controlled companies in shareholding networks.
- *Identification of roots and their outflows* is done in an analogical way.
- *Analysis of the type of relationship between a selected pair of nodes* uses firstly identification of weakly connected components for finding whether two nodes are connected to each other. If they are connected (i.e. they are in the same connected component of a graph) then identification of the type of relationship is undertaken. We use breadth-first search from and into the both nodes in order to find out whether a) node one is parent or child of node two, or b) nodes have common children or parent nodes, or c) none of both. The type of relationship is displayed to the user and the relevant path between the nodes is highlighted (see Figure 3.19).

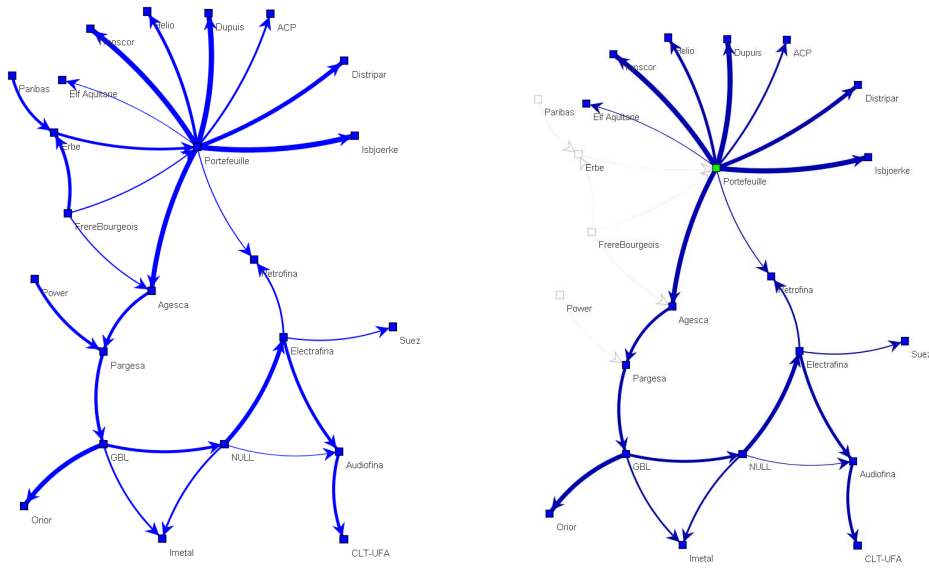


Figure 3.17.: Visual analysis of outgoing (incoming) relations. Left: Original graph. Right: Highlighting of outgoing relations of a selected graph node.

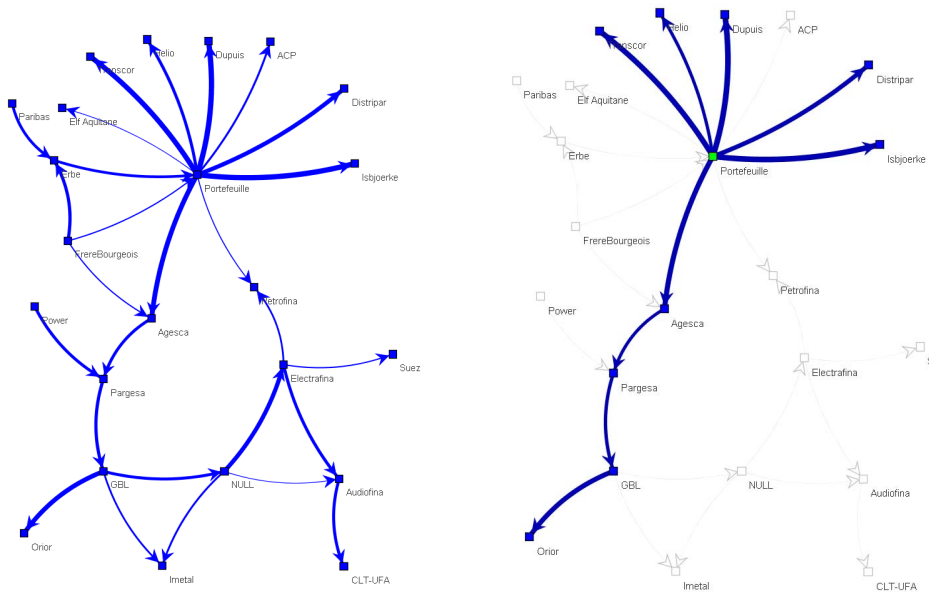


Figure 3.18.: Visual analysis of weighted/filtered outgoing relations. In this case weight of greater or equal 0.5 was used. Left: Original graph. Right: Highlighting of weighted outgoing relations of a selected node.

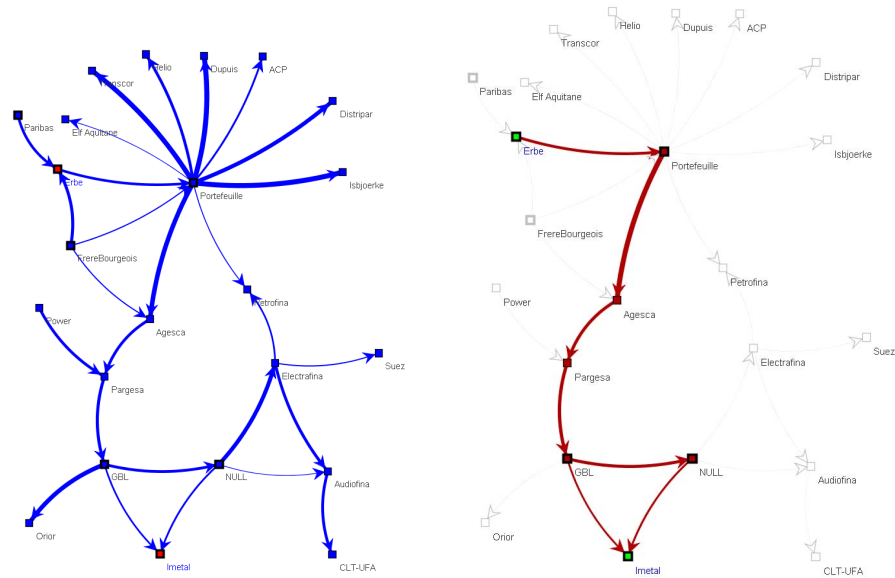


Figure 3.19.: Visualization of the relationship between two selected nodes. Left: Original graph with highlighting of the two nodes (Erbe and Imetal). Right: Highlighted relationship between the two nodes (Erbe is a predecessor node of Imetal).

3.5. Visual Analysis of Graph Motifs

3.5.1. Introduction

In this section, we present novel approaches for visual analysis of weighted directed graphs based on motif search and visualization. Description of graph motifs is presented in Subsection 3.5.2. Following, the details on three ways of visual motif analysis are described in Subsections 3.5.3, 3.5.4 and 3.5.5.

3.5.2. Graph Motifs

Motifs are interesting graph substructures (patterns) as they often carry functional information for different graph classes or occur with a high frequency in the network [MSOI*02, SS04]. In general, motifs having a functional (analytical), or semantic meaning in a particular application, those occurring with higher frequency than it would be expected for random graphs, or occur rarely are of interest. For example, the so called “feedback motif” has information filtering properties in biologic networks.

The space of possible motifs of a certain size (according to number of vertices) in directed graphs contains all possible connected graphs spanning over these number of vertices that are not isomorphic to each other [SS04]. Their number rises exponentially with motif size [Cal07]. The number of relevant substructures is, however, smaller and is dependent on the specific task and application. Nevertheless, there are several structures that are of common interest to multiple scenarios.

In our work, we first identified a set of possible relevant motifs common to various applications (see Figure 3.20). Our selection of predefined motifs has been motivated by these observations: We focused on motifs that are often overrepresented in biological, social or financial networks and are also supposed to reveal important information. In shareholding networks, certain company structures (i.e., graph structures) are used or company controlling (e.g., caro, feed-forward) or “(outgoing)-star motif” highlights big “shareholding companies”. Explanation of these motifs specifically for shareholding networks is provided in Section 3.7.4.1. The presented motifs may not cover all analysis tasks and therefore, in our work these motifs can be interactively extended by user defined structures on demand using the visual interactive interface illustrated in the Figure 3.22 (center).

In general, motifs can be parametrized by posing constraints on the values of edge or node attributes (e.g., edge weight or node label). These parameters are used for filtering of interesting motifs according to their properties. For example, in weighted graphs, minimum, maximum, total or average weight in the motif can be used as a filter threshold. In shareholding networks, the sum of all shares in a company greater than 50 % creates controlling power, therefore possibly only motifs over this threshold may be of interest.

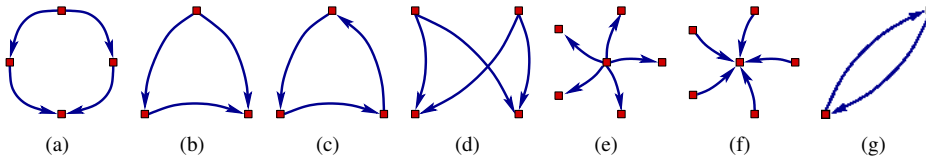


Figure 3.20.: Selected graph motifs. a) Caro, b) Feed-forward, c) Feedback, d) Double cross, e) Out-star, f) In-star and g) Reciprocity.

The algorithmic determination of motif frequencies (i.e. maximum number of occurrences of a motif in a graph) for all possible motifs of a certain size is a NP-hard problem [GK07]. Schreiber et al. [SS04] define three concepts for determination of pattern frequency: Using concept F_1 , all occurrences of a pattern are counted. In concept F_2 , only motifs with non-overlapping edges are counted. Concept F_3 restricts F_2 with non-reuse of nodes in counting. Concept F_1 finds all possible matches of a motif, thereby it shows a complete overview of patterns in the graph. Therefore, we use this concept in our work.

For finding motifs, in general, exact search is preferred [SS05, WR06, MSOI*02]. However, in many approaches heuristics are used in order to accelerate the analysis [SS04, Sch08, Wer06, GK07]. These heuristics are developed for finding all motifs of a certain size. In our case, we concentrate on one selected (predefined or user-defined) motif. Therefore, we follow approach from [GK07] for finding motifs.

In our work, we have implemented specific search procedures for the predefined motifs in order to reduce their computational time. They use motif structure for search and apply the symmetry breaking rules for avoiding isomorphic matching as proposed in [GK07] (see Algorithm 3.5.2.1 for an example feed-back search using notation presented below). Such adjustments are not possible for user-defined motifs that are not included in the set of predefined motifs, as their structure is unknown in advance. In the following, we therefore present a modified algorithm for finding user-defined motifs based on the algorithm of Grochow and Kellis [GK07].

Algorithm 3.5.2.1 FINDFEEDBACKMOTIFS(G)

Input: $G = (V^G, E^G)$ – the searched graph, all vertices $v^G \in V^G$ have unique identifiers $ID(v^G) \in \{1, \dots, n\}, n = |V^G|$

Output: M – all matches of feed-back motif found in graph G . $M = \bigcup M^i = (V^i, E^i), V^i \subseteq V^G, E^i \subseteq E^G$

```

 $M \leftarrow \emptyset$ 
for all  $v_1^G \in V^G$  do
  for all  $v_2^G \in (Successors(v_1^G) - v_1^G)$  do
    if  $ID(v_2^G) > ID(v_1^G)$  then {Symmetry breaking}
      for all  $v_3^G \in (Successors(v_2^G) - v_2^G)$  do
        if  $ID(v_3^G) > ID(v_1^G)$  then {Symmetry breaking}
           $e_{1,2}^G \leftarrow (g_1^G, g_2^G)$ 
           $e_{2,3}^G \leftarrow (g_2^G, g_3^G)$ 
           $e_{3,1}^G \leftarrow (g_3^G, g_1^G)$ 
           $M \leftarrow M \cup (\{v_1^G, v_2^G, v_3^G\}, \{e_{1,2}^G, e_{2,3}^G, e_{3,1}^G\})$ 
        end if
      end for
    end if
  end for
end for
return  $M$ 

```

The algorithm of Grochow et al. [GK07] describes motif search in undirected graphs. For finding motifs in directed graphs, this algorithm is performed first and then the found matches are divided according to their isomorphic types. This part of the algorithm is however not described in more detail. Using motif search first in undirected algorithms increases the need for additional discrimination of found motifs according to the edge direction. For example, in undirected case, feed-forward and feed-back motifs are not uniquely distinguished. So more possible matches need to be checked for the match. We overcome this drawback by directly checking for edge direction during the algorithm iterations leading to lower number of matches that need to be checked for

isomorphism at the end of the procedure and earlier termination of the iterations owing to stricter checking criteria for match of the motif. Moreover, these direction-preserving checks also include checking for bi-directional edges, which is not possible in the algorithm of Grochow et al. [GK07]. In practice, checking for neighbors and degrees in the original algorithms has been replaced by checking for children and parent relationships and in- and out-degrees. Additional improvements include 1) sorting of motif vertices only once at the beginning of the procedure not in each iteration, 2) omitting of the checks for non-neighboring nodes in the ISOMORPHICEXTENSIONS (see Algorithm 3.5.2.3), as they do not lead to new restrictions, and 3) speeding up the finding of new node matches by checking only those nodes that are neighbors to the already matched node being examined (not all matched nodes) and by checking for support of the new match. These changes lead to significant improvement in computational time. The improvements depend on the structure and size of the graph. For example, for finding feed-backward motifs in a graph with ca. 105,000 nodes and ca. 124,000 edges (see Section 3.7.2), our procedure is ca. 10 times faster using comparable implementation of the algorithm.

The new algorithm contains the procedures described below (see also Algorithms 3.5.2.2 to 3.5.2.4). The main procedure is FINDSUBGRAPHINSTANCES (see Algorithm 3.5.2.2). In these algorithms, if possible, we use the same notation as in [GK07] in order to ensure an easy comparability of the two algorithms. The algorithm searches for all matches M of motif graph $H = (V^H, E^H)$ found in graph $G = (V^G, E^G)$. V^H and V^G represent the set of vertices of graph H and G . E^H and E^G are sets of edges of graphs H and G . $M = \bigcup M^i = (V^i, E^i)$, $V^i \subseteq V^G$, $E^i \subseteq E^G$, V^i are determined by the function $f_V^i : V^H \rightarrow V^G$, and E^i by $f_E^i : E^H \rightarrow E^G$. Note that superscript denotes the graph, whose elements are referenced to. In the algorithms, m^H , dm^H , and m^H , dm^H denote vertices of graph H and G respectively. $D_V^H \subseteq V^H$ represents the domain of function f_V and $D_E^H \subseteq E^H$ represents the domain of function f_E . Calling of a procedure is represented by small capital letters, e.g., CALLPROCEDURE.

- FINDSUBGRAPHINSTANCES(H, G): The main procedure for finding all matches of the motif H in graph G . It follows Grochow et al. [GK07]. All nodes $g^G \in V^G$ are iterated trying to be matched them one of nodes $h^H \in H$. At the end of each iteration, this node and its adjacent edges are removed from the graph (see Algorithm 3.5.2.2).
- FINDAUTOMORPHISMS(H) the procedure finds all automorphisms of the motif H . For finding automorphisms, FINDSUBGRAPHINSTANCES(H, H) is called without using symmetry breaking parts.
- FINDSYMMETRYBREAKINGCONDITIONS(H, Aut^H) The procedure ceates symmetry breaking conditions for the motif graph H given all its automorphisms A^H . The procedure follows the same procedure proposed by Grochow et al. [GK07]. We refer to this publication for details.
- ORDERNODES(G): A procedure for sorting the nodes V^G of graph G in increasing order using criteria, which take into consideration the in- and out-degrees of nodes (in contrast to using total degrees as in Grochow et al. [GK07]). The checks start with comparing the larger of out- and in-degree. In case of maximum degree equality, the other out-/in-degrees are compared. If they are equal, the largest out- and in-degree of all neighbors of the two vertices are used for comparing them in the same way.
- NODEGCANSUPPORTNODEH(h^H, g^G, H, G): The procedure checks whether the node h^H can support the node g^G . The test is successful, if both the out-degree and in-degree of node g^G are larger or equal to the corresponding out-/in-degree of node h^H .
- ISOMORPHICEXTENSIONS($f_V, H, G, [C]$): A recursive procedure finding new motif matches via adding new node matches $f(m^H) = m^G$ to the already matched motif nodes in f_V . The procedure is terminated when all motif nodes V^H have a match in graph G (see Algorithm 3.5.2.3). Parameter C is used when symmetry breaking conditions are to be applied.
- TESTFORFULLMATCH(f_V, H, G): Procedure testing if the matched nodes, if all vertices of H have corresponding vertices and edges in G (see Algorithm 3.5.2.4).

- **GETMOSTCONSTRAINEDNODENOTIND**(f_V, H): Procedure for finding new node $m^H \notin D^H$ from motif graph H to be matched in graph G as the most constrained node $m^H \notin D^H$ from a set of all neighbors of the already matched nodes $d \in D^H$, $D^H \subseteq V^H$. The procedure follows Grochow et al. [GK07]. Additionally, a node $dm^H \in D^H$ is found, which is the corresponding neighbor of $m^H \notin D^H$.
- **GETPOSSIBLEMATCHESOFMING**($dm^H, m^H, dm^G, f_V, H, G$): The procedure finds all possible matching nodes m^G of node m^H in graph G . The difference to Grochow et al. [GK07] is the iteration part. They iterate over all nodes neighboring to all nodes in D that are not in D , not just those that are neighbors to the already matched node dm^G that is currently being matched. We also differentiate according to the type of relation of m and dm , whether we search for successors, predecessors or both (in case of bi-directional relationship). We additionally check for support of the new match (see Algorithm 3.5.2.5).
- **TESTPOSSIBLEMATCH**($m^H, m^G, f_V, H, G, [C]$): Procedure testing whether the found possible matching node $m^G \in G$ of node $m^H \in H$ using **GETPOSSIBLEMATCHESOFMING** obeys the neighboring and possibly also symmetry breaking constraints. In comparison to Grochow et al. [GK07], successor and predecessor nodes (not neighbors in general) are tested independently. Testing for non-neighbors is omitted (see Algorithm 3.5.2.6).

Algorithm 3.5.2.2 FINDSUBGRAPHINSTANCES(H, G)

Input: $H = (V^H, E^H)$ – the motif graph, $G = (V^G, E^G)$ – the searched graph

Output: M – all matches of motif H found in G . $M = \bigcup M^i = (V^i, E^i)$, $V^i \subseteq V^G$, $E^i \subseteq E^G$, V^i are determined by the function $f_V^i : V^H \rightarrow V^G$, and E^i by $f_E^i : E^H \rightarrow E^G$

$M \leftarrow \emptyset$

$D_V^H \leftarrow \emptyset$

$D_E^H \leftarrow \emptyset$

$[Aut^H \leftarrow \text{FINDAUTOMORPHISMS}(H), \text{when using symmetry breaking}]$

$[C \leftarrow \text{FINDSYMMETRYBREAKINGCONDITIONS}(H, Aut^H) \text{ when using symmetry breaking}]$

$V^G \leftarrow \text{ORDERNODES}(G)$

$V^H \leftarrow \text{ORDERNODES}(H)$

for all $g^G \in V^G$ **do**

for all $h^H \in V^H$ such that **NODEGCANSUPPORTNODEH**(h^H, g^G, H, G) **do**

$f_V(h^H) = g^G$, $D^H = h^H$

$M \leftarrow M \cup \text{ISOMORPHICEXTENSIONS}(f_V, H, G, C)$

end for

$E^G \leftarrow E^G - \{e^G(g^G)\}$ {remove all edges incident to vertex g^G from E^G }

$V^G \leftarrow V^G - \{g^G\}$ {remove vertex g^G from V^G }

end for

return M

Algorithm 3.5.2.3 ISOMORPHICEXTENSIONS(f_V, H, G, C)

Input: $H = (V^H, E^H)$, $G = (V^G, E^G)$, $f_V : D_V^H \rightarrow R_V^G$, $D_V^H \subseteq V^H$, $R_V^G \subseteq V^G$, C refers to the symmetry breaking conditions to be satisfied

Output: M – a set of matches, $M = (f_V, f_E)$, where $f_E : D_E \rightarrow R_E$, $D_E^H \subseteq E^H$, $\{R_E^G \subseteq E^G\}$

```

 $M \leftarrow \emptyset$ 
 $(c, f_E) \leftarrow \text{TESTFORFULLMATCH}(f_V, H, G)$ 
if  $c = \text{true}$  and  $(f_V, f_E) \notin M$  then
     $M \leftarrow (f_V, f_E)$ 
else if  $|D| < |V^H|$  then
     $\{dm^H, m^H\} \leftarrow \text{GETMOSTCONSTRAINEDNODENOTIND}(f_V, H)$ 
    if  $\exists m$  then
         $dm^G \leftarrow f_V(dm^H)$ 
         $R^G \leftarrow \text{GETPOSSIBLEMATCHESOFMING}(dm^H, m^H, dm^G, f_V, H, G), R^G \subseteq V^G$ 
        if  $R^G \neq \emptyset$  then
            for all  $m^G \in R^G$  do
                if  $\text{TESTPOSSIBLEMATCH}(m^H, m^G, f_V, H, G, C)$  is true then
                     $f_V \leftarrow f_V \cup f(m^H) = m^G$ 
                     $\text{ISOMORPHICEXTENSIONS}(f_V, H, G)$ 
                end if
            end for
        end if
    end if
end if
return  $M$ 

```

Algorithm 3.5.2.4 TESTFORFULLMATCH(f_V, H, G)

Input: $H = (V^H, E^H)$, $G = (V^G, E^G)$, $f_V : D_V^H \rightarrow R_V^G$, $D_V^H \subseteq V^H$, $R_V^G \subseteq V^G$

Output: (c, f_E) – c and a set of edge matches f_E . c is true if a match has been found, else c is false, $f_E : D_E^H \rightarrow R_E^G$, $D_E^H \subseteq E^H$, $R_E^G \subseteq E^G$

```

 $f_E = \emptyset$ 
if  $|D_V^H| < |V^H|$  then
    return (false,  $\emptyset$ )
else if  $|D_V^H| = |V^H|$  then
    for all  $e^H = (s_1^H, s_2^H) \in H$  do
        if  $\exists e^G = (f_V(s_1^H), f_V(s_2^H))$  then
             $f_E \leftarrow f_E \cup \{e^G = f(e^H)\}$ 
        else
            return (false,  $\emptyset$ )
        end if
    end for
    if  $|R_E^G| = |E^H|$  then
        return (true,  $f_E$ )
    end if
end if

```

Algorithm 3.5.2.5 GETPOSSIBLEMATCHESOFMING($dm^H, m^H, dm^G, f_V, H, G$)

Input: $H = (V^H, E^H)$, $G = (V^G, E^G)$, $f_V : D_V^H \rightarrow R_V^G$, $D_V \subseteq V^H$, $R_V^G \subseteq V^G$, nodes dm^H, m^H, dm^G

Output: $R^G \subseteq V^G$ –

```

 $R^G \leftarrow \emptyset$ 
if  $\exists e^H = (m^H, dm^H)$  and  $\exists e^H = (dm^H, m^H)$  then
  for all  $d \in ((\text{Successors}(dm^G) \cap \text{Predecessors}(dm^G)) - R_V^G)$  do
    if NODEGCANSUPPORTNODEH( $m^H, d, H, G$ ) then
       $R^G \leftarrow R^G \cup \{d\}$ 
    end if
  end for
else if  $\exists e^H = (m^H, dm^H)$  then
  for all  $d \in (\text{Successors}(dm^G) - R_V^G)$  do
    if NODEGCANSUPPORTNODEH( $m^H, d, H, G$ ) then
       $R^G \leftarrow R^G \cup \{d\}$ 
    end if
  end for
else  $\{\exists e^H = (dm^H, m^H)\}$ 
  for all  $d \in (\text{Predecessors}(dm^G) - R_V)$  do
    if NODEGCANSUPPORTNODEH( $m^H, d, H, G$ ) then
       $R^G \leftarrow R^G \cup \{d\}$ 
    end if
  end for
end if
return  $R^G$ 

```

Algorithm 3.5.2.6 TESTPOSSIBLEMATCH(m^H, m^G, f_V, H, G, C)

Input: m^G a possible match of m^H in G , $H = (V^H, E^H)$, $G = (V^G, E^G)$, $f_V : D_V \rightarrow R_V$, $D_V^H \subseteq V^H$, $R_V^G \subseteq V^G$, C set of symmetry breaking rules

Output: true, if m^G is a match of m^H , else false

```

for all  $successor \in \text{Successors}(m)$  do
  if  $successor \in D_V^H$  and  $f_V(successor) \notin \text{Successors}(m^G)$  then
    return false
  end if
end for
for all  $predecessor \in \text{Predecessors}(m)$  do
  if  $predecessor \in D_V^H$  and  $f_V(predecessor) \notin \text{Predecessors}(m^G)$  then
    return false
  end if
end for
if  $C \neq \emptyset$  then
  return result of the check for symmetry breaking using  $C$  for the new node  $m^G$  (according to [GK07])
else
  return true
end if

```

3.5.3. Interactive Definition and Visualization of Motifs

The visual analysis of graph motifs usually employs search for a predefined motif in the whole graph and subsequent visualization of found motifs using highlighting in the original graph. In our approach, we extend these approaches. We employ search for all occurrences of a specified motif (predefined and user-defined) either in the whole graph, or in a local area around a focused node or edge from the graph neighborhood/connectivity view. The first case gives a general overview of motif frequencies and distributions in the network (being computationally more intensive) and the second one allows for the analysis of structures in which a specific node or edge is involved (the latter being more focused and less computationally intensive). The motif type can be selected from a set of predefined motifs or interactively specified using a visual graph editor (see Figure 3.21 and 3.22). Furthermore, the set of motifs that have been found can be filtered in order to focus on structures obeying certain constraints (see Figure 3.23).

The visualization of the detected graph structures employs inter alia highlighting of the located motifs in a network². It shows the found motifs in the context of the whole graph or focuses on the found motifs only. In general, simple highlighting of these graph motifs may lead to overloaded views owing to high motif occurrence or motif overlap. This is particularly the case, when many motifs are found or motifs overlap strongly (one edge or node is a member of several motifs).

The motif occlusion is therefore in our work addressed either by highlighting of a motif including a selected graph entity or motif filtering. In the first case, a selection of a node or edge shows all motifs that include the selected element. It can be used for detailed analysis of particular entities in the graph. In the second case, filtering of motifs using attribute constraints is employed. For example, motifs which include only edges with weight larger than a certain threshold can be filtered (see Figure 3.23).

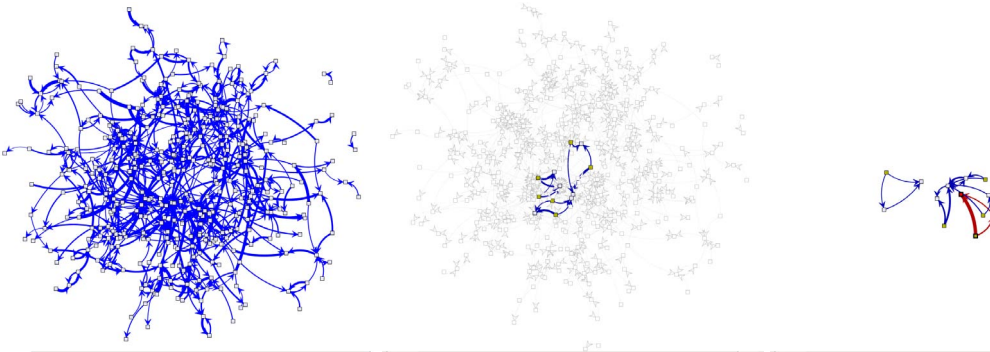


Figure 3.21.: Search and visualization of selected predefined motifs in a graph. Left: Original graph. Center: Caro motif highlighting in the context of the original graph allowing for localization of found motifs in the original graph. Right: Feed-forward motif without context with highlighting of a motif included in a node allowing for stronger focus on found graph parts.

²Graph visualization uses node-link diagrams and user-selected graph layout (from JUNG library) thereby building on a flexible choice and extension of layouts used.

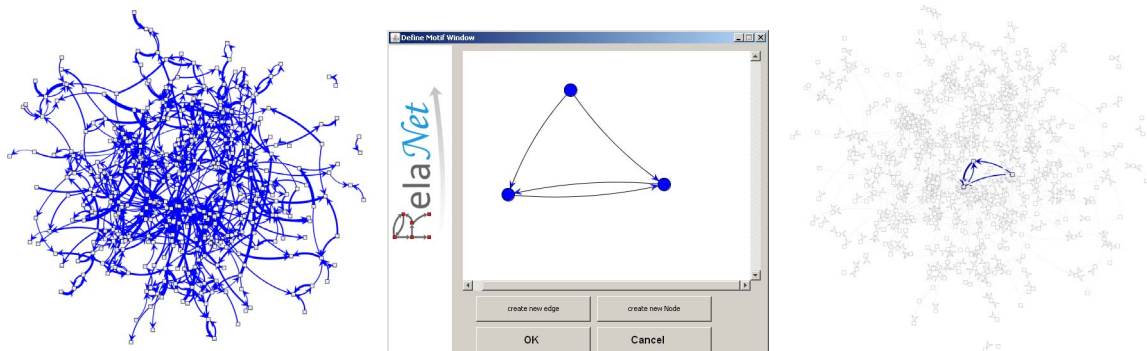


Figure 3.22.: Search and visualization of user-defined motifs. Left: Original graph. Center: Interactive motif definition. Right: Visualization of the found user-defined motifs in context of the original graph.

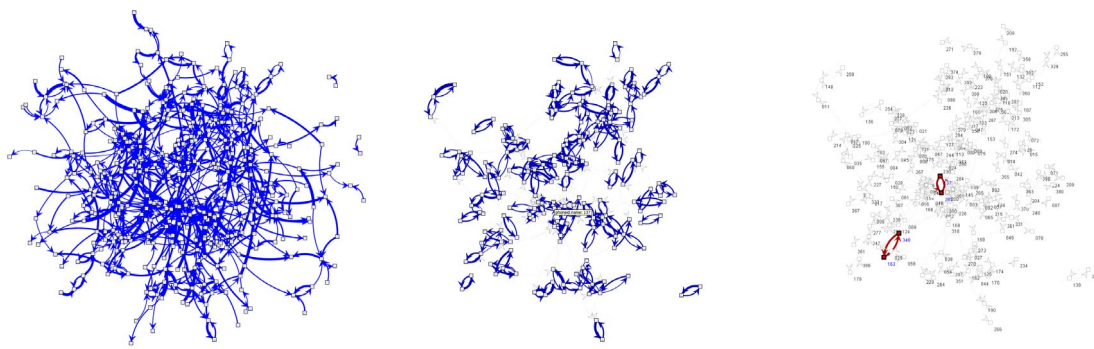


Figure 3.23.: Motif filtering example. Left: Original graph. Center: All occurrences of reciprocity motif. Right: Occurrences of reciprocity motif for edge weights larger than a given threshold.

3.5.4. Visual Analysis of Graph Changes using Motifs

3.5.4.1. Graph changes

The graph changes can be data or user driven. In the data driven case, the graph changes are loaded from a data source, in the other case, the user performs manual changes to the graph using a graphic interface. In the graph, vertices and edges may be deleted, added or modified (attribute values are changed). In the following, we focus on user defined graph changes (used in “what-if-analysis”) and the visual analysis of its impacts.

When manually applying graph changes, a change in any particular node and edge deletion may imply further needs for modifications to the network (see Figure 3.24).

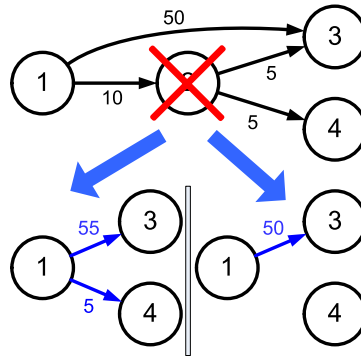


Figure 3.24.: An example of two possible implications of node deletion in a network.

In case of deleting a vertex, for example, its adjacent edges need to be deleted or redirected, and, possibly, attribute values of the new/other edges may need to be altered. These implications may not be trivial to solve automatically without losing semantic information. In particular, it is not clear whether the edges adjacent to the deleted node should be deleted (the indirect connection between nodes gets lost) or redirected (the indirect connection stays). Moreover, constraints on the edge values (e.g., weights corresponding to shares in shareholding networks) can be posed and need to be taken into account when processing graph changes automatically. For example, in case a company does not exist any more (a node is deleted) its shares in other companies need to be redistributed so that the sum of shares in each company equals 100%. In our work, we use a semi-automatic change management. The computer identifies the need for indirect changes and shows a proposal for the edge redirection, which can be accepted or altered by the user (see Figure 3.25).

3.5.4.2. Interactive Visualization of Impacts of Graph Changes

After performing user-defined graph change, the modified entities (e.g., new edges) are locally searched for occurrence of motifs for identification of structural changes in the network. For example, by adding an edge, a new shareholding relationship between two companies is created, which may lead to new controlling powers (see Figure 3.26).

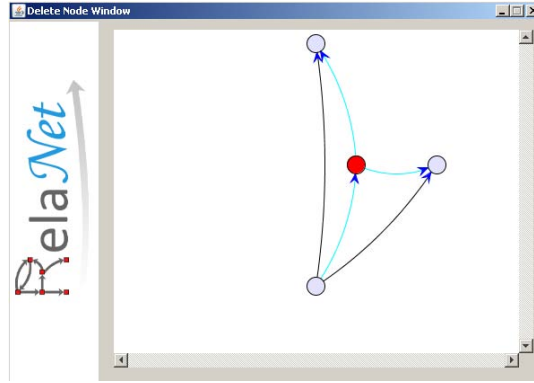


Figure 3.25.: Proposal for edge redirection after deleting a node. The original edges adjacent to the deleted node are shown in light blue color and the proposed new edges are colored in black.

3.5.5. Visual Analysis of Graphs Using Motif-based Graph Aggregation

3.5.5.1. Graph Aggregation

The *graph aggregation* merges a defined group of nodes into a single aggregated (or merged) node. The edges between the nodes in the group are removed and the edges between in the nodes which should be aggregated and nodes in the rest of the graph are merged. All edges from each node outside the group to all nodes in the group (incoming edges) are merged into one edge per each node outside the group. Similarly the out-going edges are aggregated. For illustration of the algorithm, see Figure 3.27. The attributes of the aggregated node are calculated based on group node attributes using interactively user-defined aggregation function (e.g., summing up number of employees in a group of aggregated companies).

Aggregation types In this thesis, we employ three *ways of defining group of nodes for aggregation*: interactively user-defined, based on node attribute values and motif-based.

- *Interactive definition* of node groups for aggregation is provided by highlighting of selected nodes. It is useful for grouping of nodes identified by the user as belonging together according to their view/expertise.
- In the *attribute-based* aggregation, the user-selected node attribute is used for node grouping. Based on cardinality of the attribute values, either the distinct values of attribute are used as criteria (e.g., sectors) or discretization is performed before grouping (e.g., binning of companies according to company size into small, middle and large companies).
- For *motif-based graph aggregation*, the graph is first searched for occurrence of a selected motif, if appropriate the results are filtered according to user-defined criteria and then all nodes included in this motif are merged (see Figure 3.28). Each aggregated node thereby includes a particular graph structure. In a simple case each found motif is aggregated into a single merged node. However motifs may overlap, i.e., one node or one edge is included in several substructures. For aggregating overlapping motifs, we can either duplicate the nodes in multiple motifs or aggregate multiple connected motifs into one node. In this thesis, we employ the latter technique.

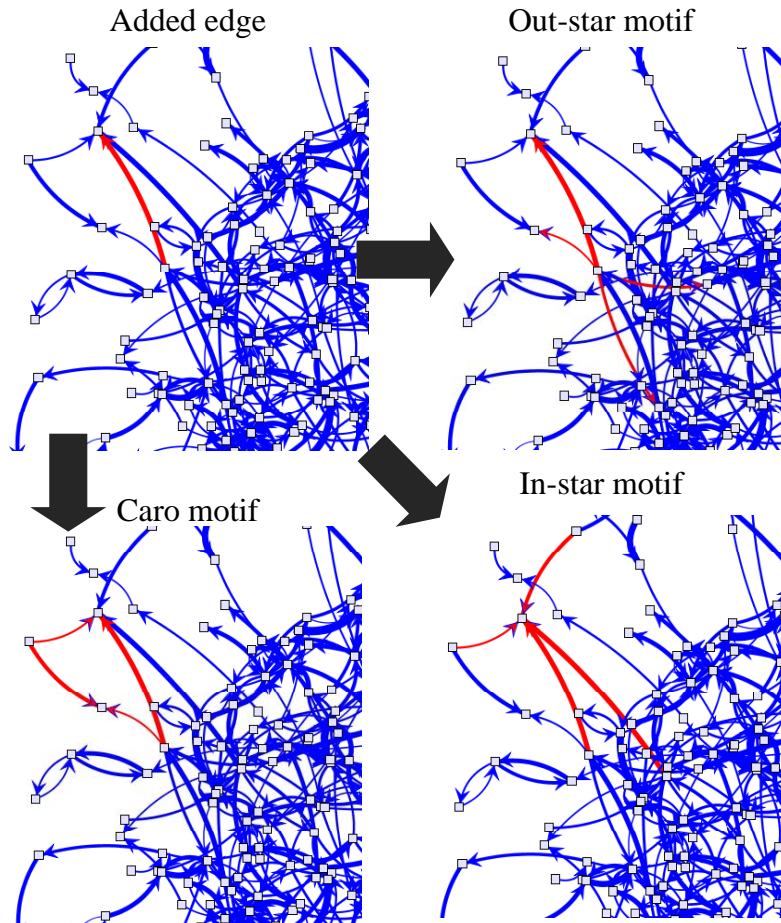


Figure 3.26.: Visual analysis of changes in local graph structures after adding an edge. The figure shows the new added edge highlighted in red (top left) and the new identified graph motifs emerging from the user-defined graph change (highlighted also in red color).

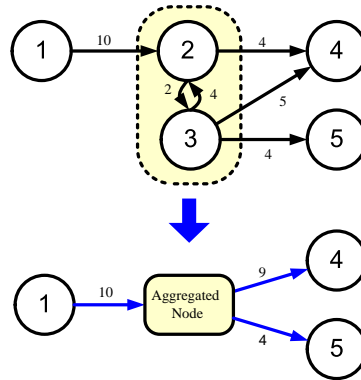


Figure 3.27.: Reciprocity motif aggregation example.

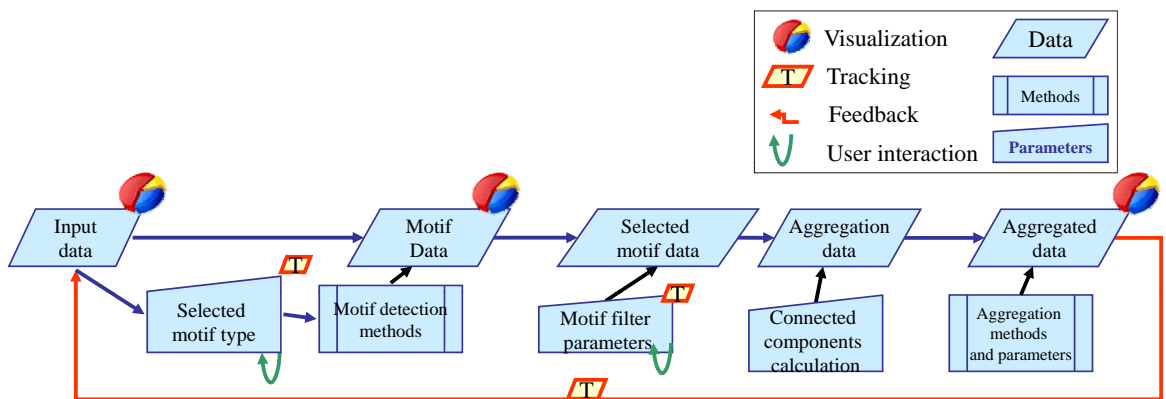


Figure 3.28.: Motif aggregation schema. Motif aggregation starts with selection of motif type, continues with detection of the selected graph, possible filtering of found motifs, creation of the connected components of the motif graph and aggregation of the motif nodes. This process can be repeated more times.

The aggregation of each group of selected nodes (by one of the three above mentioned approaches) is merged into individual aggregated node (see Figure 3.27). The edges between nodes within the group and outside the group are aggregated as well.

The aggregation *process can be successively repeated*. The type and parameters of aggregation (i.e., motifs, user-defined or attributes and their characteristics) can stay constant or can be changed in each step on demand. In each iteration, a more simplified graph (a new aggregation layer) is created. This is especially useful for very large graphs, as the multiple successive aggregation may reveal “higher level” structures in the original graph (see Figure 3.33). For example, in analysis of economy sectors, the relationships at several levels of sectoral membership can be examined. It needs to be noted that the order of aggregation types used for aggregation plays a significant role in the structure of the output graph. Different orders may lead to different results. Therefore, it is useful to track the sequence of the aggregation steps for later reconstruction or change of the aggregations steps (see Figure 3.29). The figure shows the graph sizes in each aggregation step and icons of types of aggregation employed and, if applicable, the number of discrete motifs used in aggregation. In addition, we offer tracking of graph features for analysis of impacts of aggregation on graph structure (see Figure 3.30). The figure shows the development of selected graph features (including number of motifs) in each step and icons of types of aggregation used. This allows for tracking of impacts of structural changes in the graph.

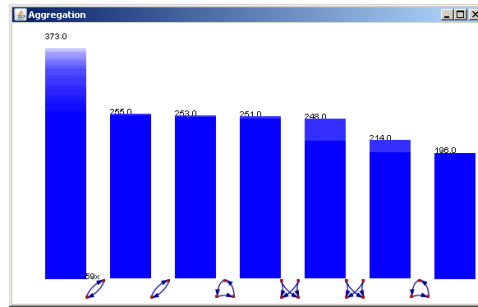


Figure 3.29.: Tracking of aggregation process showing graph size (bars) and type of aggregation (icons) in each aggregation step in a sequence.

3.5.5.2. Interactive Visualization of Graph Aggregation

In our system, the aggregated nodes/edges visually completely replace the original nodes and edges. The advantage is that the aggregated graph is simpler and therefore better readable. However, it is not possible to see the original graph structure in the aggregated one directly. For this purpose, we offer the option to show icons in the node center to indicate the motif type used for the aggregation (see Figure 3.32) and a detailed view on the aggregation hierarchy for a selected node (see Figure 3.31).

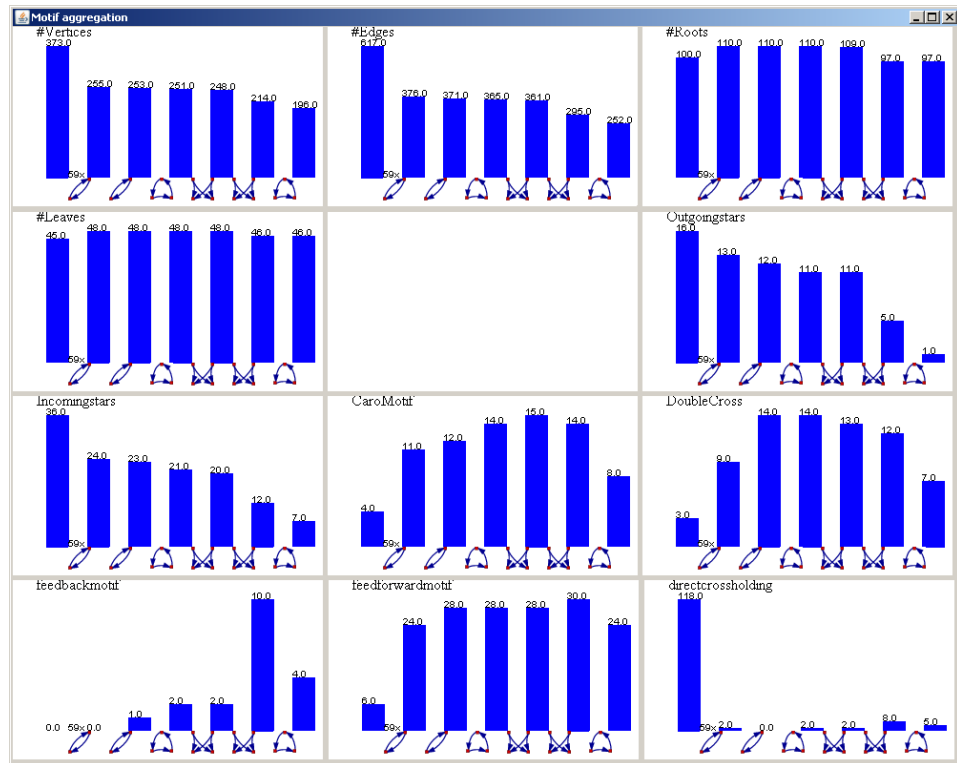


Figure 3.30.: Tracking of the aggregation process showing values of graph features in the order of aggregation steps. In the lower part the type of each aggregation step is displayed using icons.

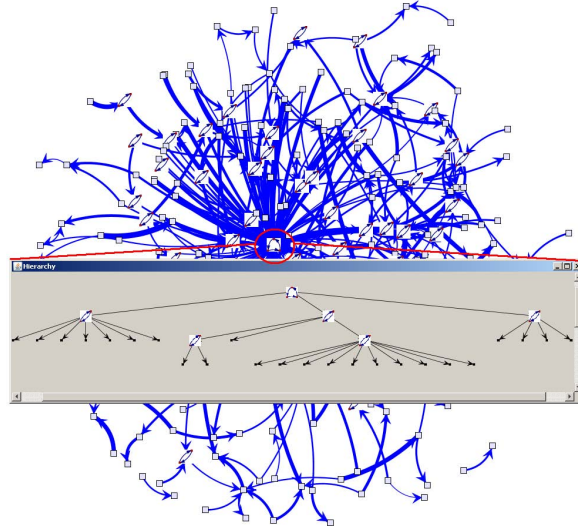


Figure 3.31.: Aggregation hierarchy for a node showing types of aggregations employed, their sequence and the nodes used in each aggregation step.

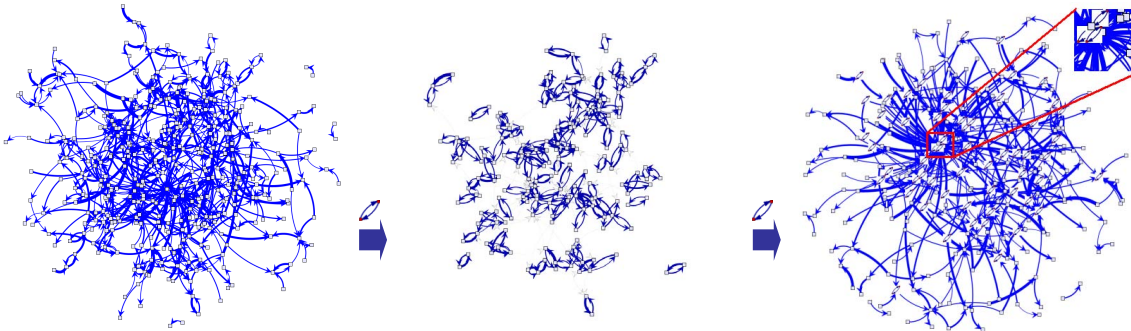


Figure 3.32.: Example of motif-based graph aggregation. Left: Original graph. Center: Reciprocity motifs selected for aggregation. Right: Aggregated graph with node icons indicating type of aggregation.

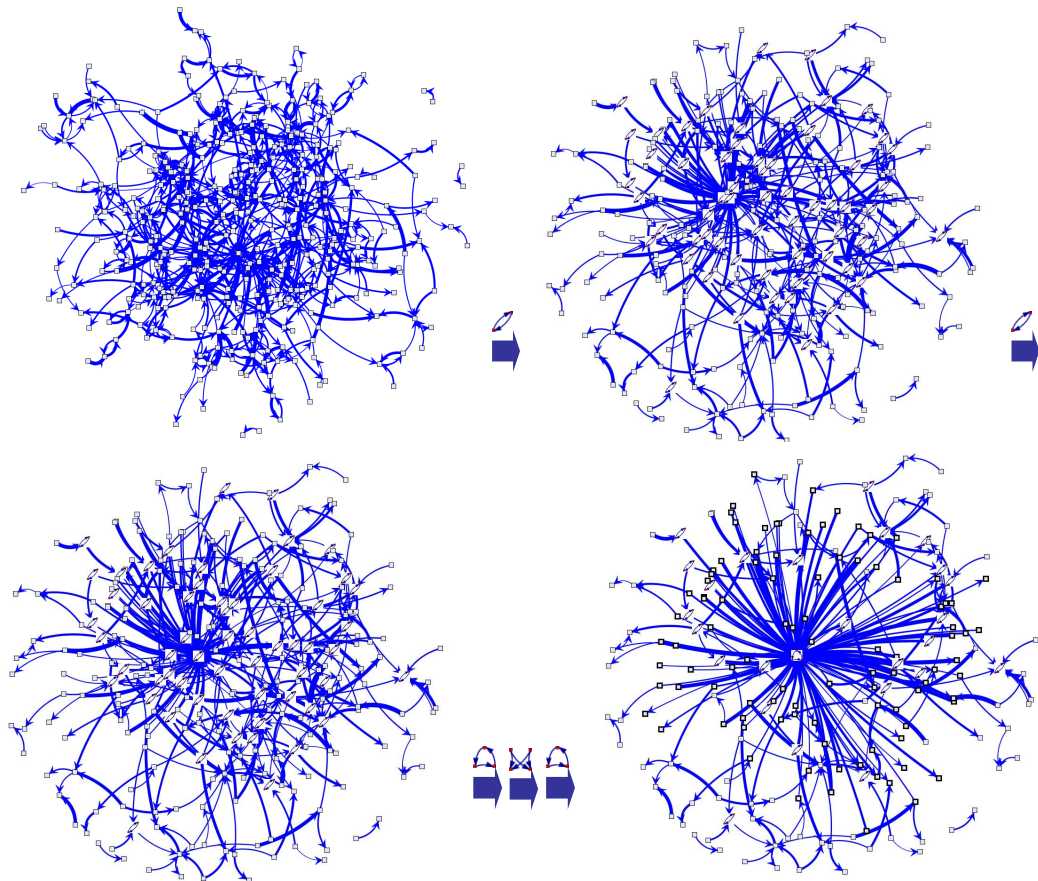


Figure 3.33.: Example of hierarchic motif-based graph aggregation with multiple aggregation steps used for graph simplification and analysis of graphs on multiple levels of abstraction.

3.6. Visual Analysis of Many Graphs Using SOM Clustering

3.6.1. Introduction

In this section, we present a novel approach for interactive visualization and exploration of graphs with many connected components based on clustering of its connected parts. The Self-Organizing Map (SOM) algorithm is used as it offers robust clustering and is well suited for visualization. The SOM cluster analysis is driven by a rich set of topology-based graph features, which can be interactively selected and combined by the user. The visual analysis system offers exploration of the graph space by interactive visualization of the feature space, clustering results and clustering quality. Furthermore it offers the possibility of saving and loading of clustering results using different clustering parameters and user-defined annotations of the process. It supports reproducibility of the analysis and insights gained.

The remainder of this section is structured as follows. Subsection 3.6.2 introduces our set of graph features we use for clustering. Subsections 3.6.3, 3.6.5 and 3.6.6 describe the main visualization and interaction results for exploration of the graph feature space, feature selection support and interactive visualization of clustering results and SOM clustering quality.

3.6.2. Graph Features for Measuring Graph Similarity

SOM clustering relies on a proper definition of similarity between entities. For graphs, two major approaches exist:

- Similarity based on transformation, such as the Edit-2 Distance for undirected acyclic graphs [ZWS96]
- Feature-based approaches, such as the graph histogram [PM99] which capture important data attributes in form of a vector or histogram. In this case, distances between data elements are calculated using vector-space or histogram distance functions.

For measuring graph similarity, we use the latter technique. We describe the graph structure by a set of appropriate graph properties (features). The selection of features, in general, depends on the type of network (directed vs. undirected, weighted vs. unweighted, with vs. without node labels, with vs. without node weights etc.). Moreover, the network semantic plays a role when defining the set of graph features to use. For instance, the sum of all weights on incoming links in shareholder networks should always sum up to 100% and is therefore not informative. In contrast, the same measure in flow networks illustrates the flow strength through the nodes.

Based on graph topology literature [Die05, Cal07, BBPS*04, Bon87, Sch08, CCR03, PM99], we chose a set of graph features referred to as important for weighted directed graphs and include also motif-based features to assess the structural patterns in graphs. We categorized our supported features into five groups: general features, degree distribution features, distance features, reciprocity features, clustering features, and motif-based features. We briefly introduce each feature set in the following. For detailed definitions of respective features in each set, we refer to the above-mentioned literature.

1. **General features** measure general properties of a network. Examples include the size of a network (number of nodes), the degree of completeness (number of links relative to the number of possible links), the average edge weight.
2. **Reciprocity features** in a directed network indicate whether node links are reciprocal, meaning that if there is a link from nodes A to B, then there exists also a link from nodes B to A. The set includes (weighted) reciprocity and the correlation coefficient of the adjacency matrix.
3. **Distance features** measure lengths of paths between nodes in a network, e.g., diameter of a graph.

4. **Clustering features** measure the probability that two nodes that are neighbors to a third node, also share a link between each other. Different measures of clustering coefficients (weighted, in/out-clustering, etc.) can be used.
5. **Degree distribution features** show the division of nodes according to their (in/out)degree. The features include average/maximum relative node degree, relative number of loops, relative number of leaves, and relative number of roots. Additionally, degree correlation/assortativity can be used.
6. **Motif-based features** measure structural properties of graphs by the frequency of certain predefined sub-structures (motifs) occurring in the given graph. For more information on graph motifs we refer to Section 3.5.2 on page 71. Figure 3.20 shows the set of motifs which we currently consider.

The introduced features do not cover all possible graph types and all graph domains and therefore can be extended according to the particular graph type and the use case. For example, for labeled networks, features describing label distribution can also be used, or in other cases, centrality features may be relevant as well. All the above-mentioned features have their strengths and weaknesses for different analysis domains. The particular set will be determined by the given analysis task and can be selected interactively (see the following section).

We note that the time complexity of the feature extraction typically depends on number of connected components, the size of each component, and the given type of feature. The calculation often can be accelerated by using parallel processing techniques. As the feature calculation is performed only once for each graph and feature, therefore also more expensive features can be considered, given the available resources.

3.6.3. Interactive Feature Selection and Visualization of Feature Space

The selection of features, their normalization and weighting influences the result of the SOM clustering. Depending on the type of network, use case and user task at hand, an appropriate combination of graph features needs to be formed. We would not like to define an optimal set of selected features (based on our experience), but offer the user the possibility to interactively select a suitable feature set (see the following paragraph) based on a particular use case.

We support interactive selection and weighting of features via the user interface depicted in Figure 3.34. A set of sliders allows to set weights for each implemented feature. The sum of weights is normalized to 1.0 and changing of a weight of a feature influences the weights of the other features so that the sum stays constant. This allows to create variable user-preferred weighting schemes.

The user interface includes a heatmap visualization of the correlation matrix of the respective features, which helps in selecting features, e.g., the most orthogonal (uncorrelated) ones. The manual feature selection is assisted by simple tools that assess the feature relevance and suggest the user to include or exclude them from the set. The weights of features with zero variance are automatically preset to zero. The features that are highly correlated with many other features are proposed to be excluded from the analysis. For the finding of the most correlated features a heuristic is used. Firstly, all pairs of features with an absolute value of correlation higher than a threshold are selected. Then from this set of features those who correlate with most number of other features are selected. If needed, more sophisticated semi-automatic feature selections (see [LM07]) could also be included.

All features are *normalized* in order to allow an easy proportional weighting. The normalization is based on graph theoretical aspects. Each feature score is calculated as a fraction of the actual value of the feature, relative to the theoretic maximum value. This yields a $[0, 1]$ -normalization, which generally gives good results in the SOM clustering according to our observation. Note that another way would be to use the expected value of each feature in random graphs for feature normalization (instead of its maximum theoretic value).

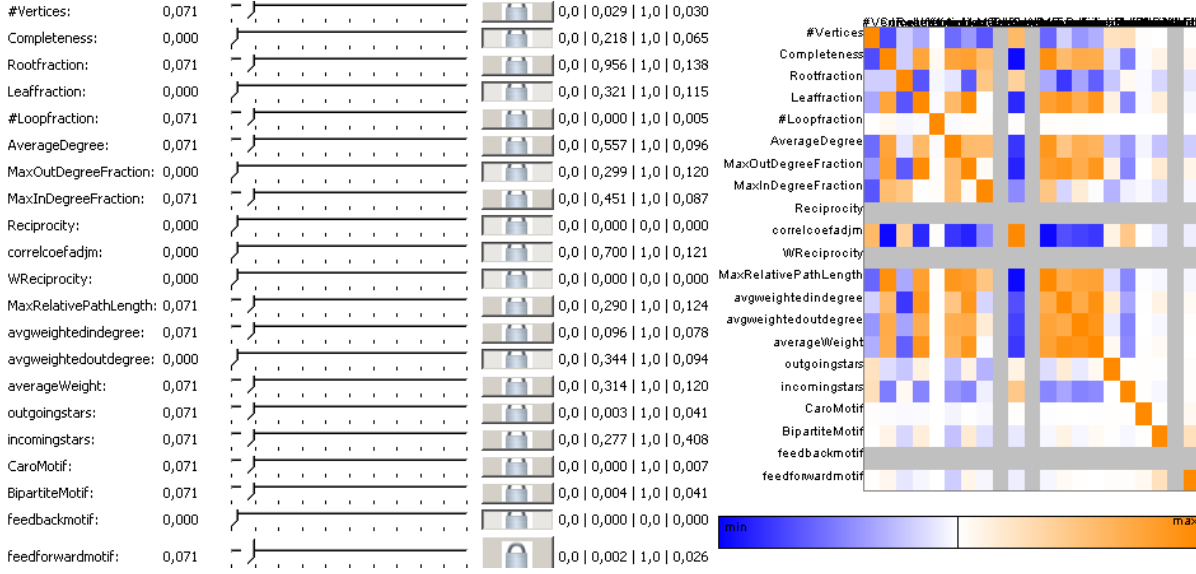


Figure 3.34.: Interactive selection and weighting of features determining the graph similarity calculation. From left to right in the picture: Feature names, current feature weights, weight adjustment interface, basic feature statistics, feature correlation matrix with color scheme from blue (-1) via white (0) to orange (1).

3.6.4. Interactive SOM Parameter Setting

Before the SOM Clustering process can start, learning parameters need to be set. The parameters include the type of initialization of the SOM grid (random or linear based on PCA of the features), the learning radius, the number of iterations and the size of SOM grid. In our work we use two stage learning process and we offer interactive user interface for adjusting the learning parameters (see Figure 3.35). The initial settings follow the recommendations by Kohonen and Vesanto [Koh01, Ves99].

3.6.5. Interactive Visualization of SOM Clustering Results

The SOM algorithm, equipped with discriminative features, usually provides meaningful results, showing an effective overview of the types of graphs in the data space. In this section, we present our approaches to visualization of the overall clustering results and exploration and refinement of individual SOM cells.

Visualization of Overall Clustering Results When visualizing the results of graph clustering, there is no direct representation of SOM prototype vectors as graphs, i.e., there is no direct function creating a graph from a presented graph feature vector which we can use to create a graph of the SOM prototype. Therefore we can either show these centers in an abstract way using multidimensional visualization techniques (such as parallel coordinates or radial plots) or use a representative graph (e.g., the nearest neighbor to the center).

In our work, we support both types of views. As default presentation, *the clustered results* are visualized by showing one representative graph for each cell on the SOM grid. The representative is chosen as the nearest

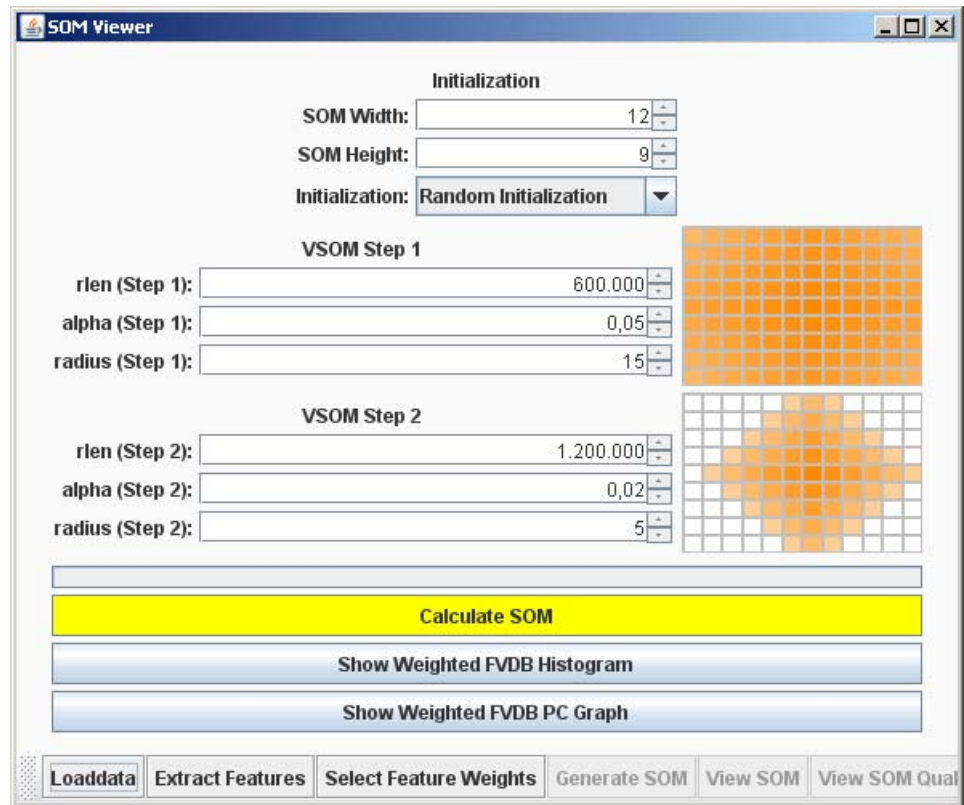


Figure 3.35.: Interactive setting of SOM learning parameters. Left: the selected parameters. Right: visualization of the SOM grid size and neighborhood function using heatmap display.

neighbor sample graph to the respective SOM prototype vector. Figure 3.36 (center) illustrates a graph cluster map. The background color of each SOM grid cell indicates the relative size of the cell, measured by the number of sample graphs matched, relative to the maximum number of samples at any SOM grid cell. For large SOM grids, the size of the grid cell can be very small which influences the readability of the displayed graphs. In this case, either the representative graph may be shown on demand in an extra view or hierarchic SOMs can be used.

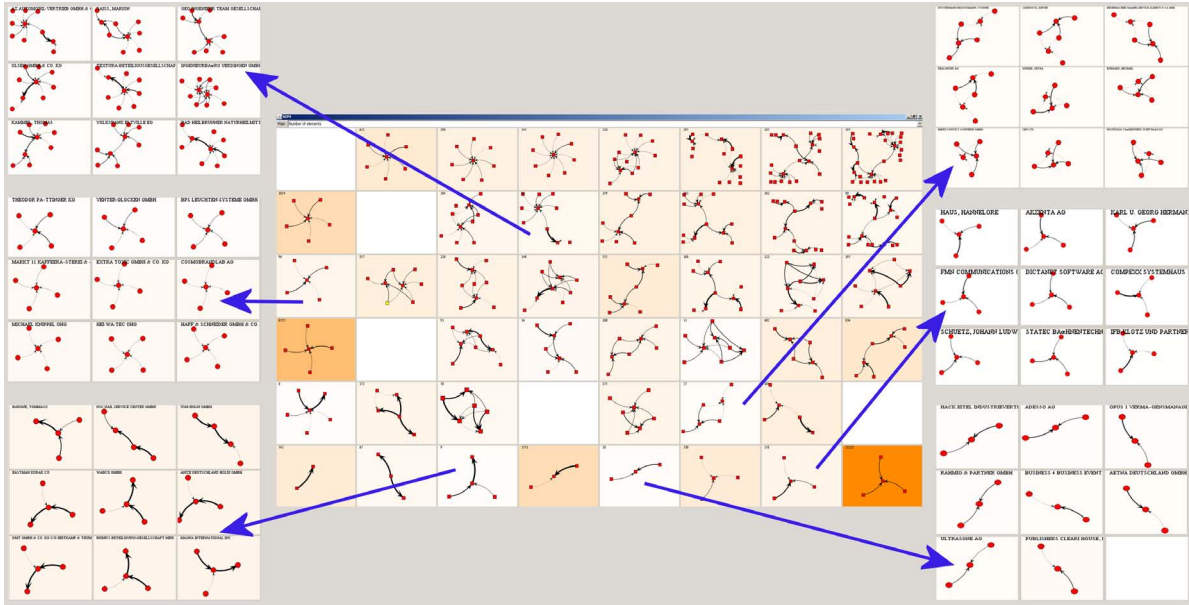


Figure 3.36.: Visualization of SOM clustering results (center). Each cell contains the nearest neighbor graph, while the background color indicates the frequency of the cell elements in each cell. Along the cluster map, several member views showing a set of nearest cell members is shown. The member views allow interactive exploration of graph clusters.

The visualization of a connected component in the grid employs node-link diagrams with edge width corresponding to the edge weight and arrows showing the edge direction (see Figure 3.36). Standard layouts provided by JUNG [OFS] were included for visualization of the graphs. The choice of the applied layout is user dependent.

The visualization of clustering results is additionally supported by display of so-called *component planes* (see Figure 3.37). This view shows the distribution of the individual features in the resulting SOM prototype vector matrix. It shows the values of each feature characterizing the cluster center across the SOM grid. The values are displayed as a heatmap. For example, the component plane for the graph size feature (top left), shows that graphs with larger number of nodes are concentrated in the right up corner of the SOM grid and smaller in the left lower corner. On demand, the values of a selected feature are displayed as the background color of the SOM grid cells (see Figure 3.38).

Exploration and refinement of SOM result In order to explore the members of individual SOM cells, the cell elements are displayed on demand in a so-called *member view*. Figure 3.36 illustrates member views of several SOM cells, arranged around the SOM grid. For cells with a large number of members (hundreds, thousands), it is possible to visualize the distance distribution or feature distribution of the cells (see the next section) and explore

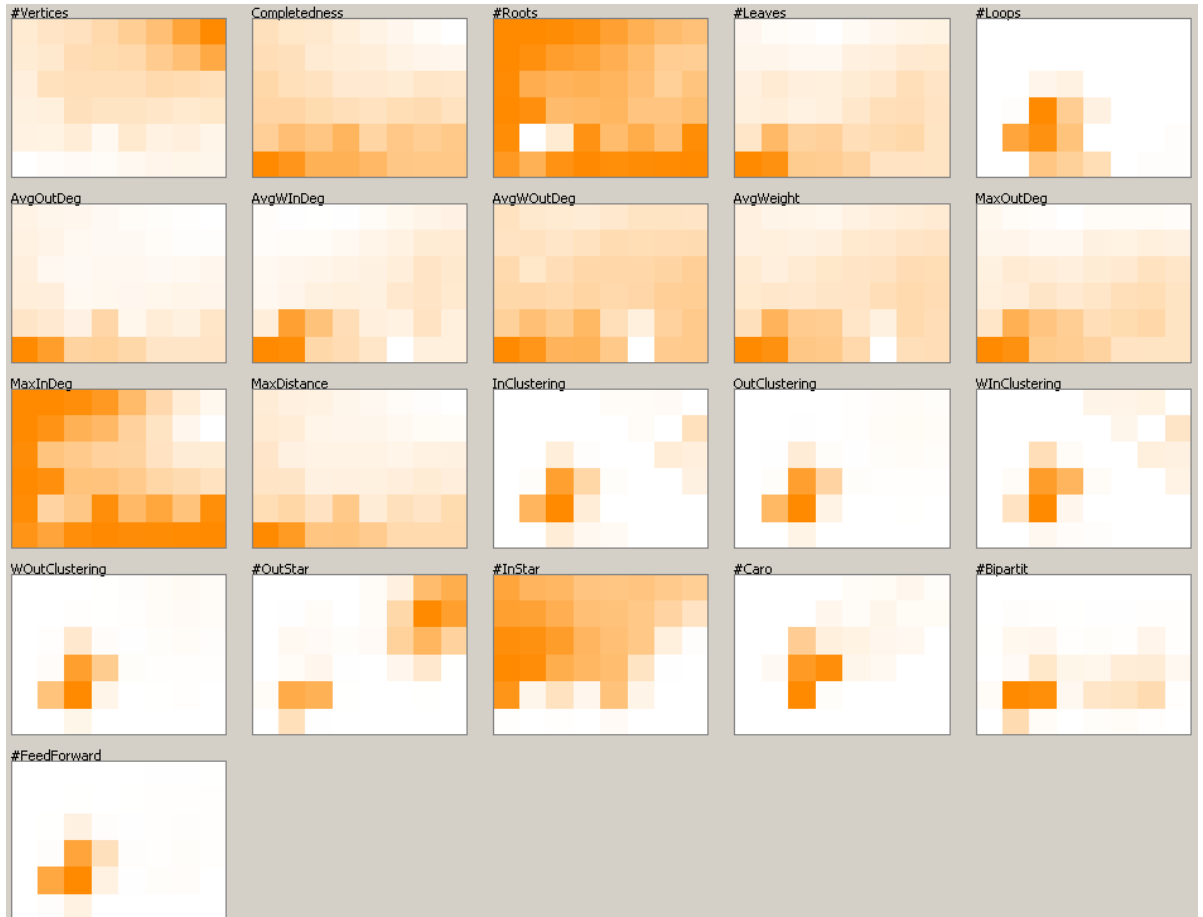


Figure 3.37.: Visualization of component planes for analysis of feature distribution across the SOM grid. It shows the values of each graph feature in the cell center across the SOM grid using a heatmap matrix. White color indicates low values and orange color means high values.

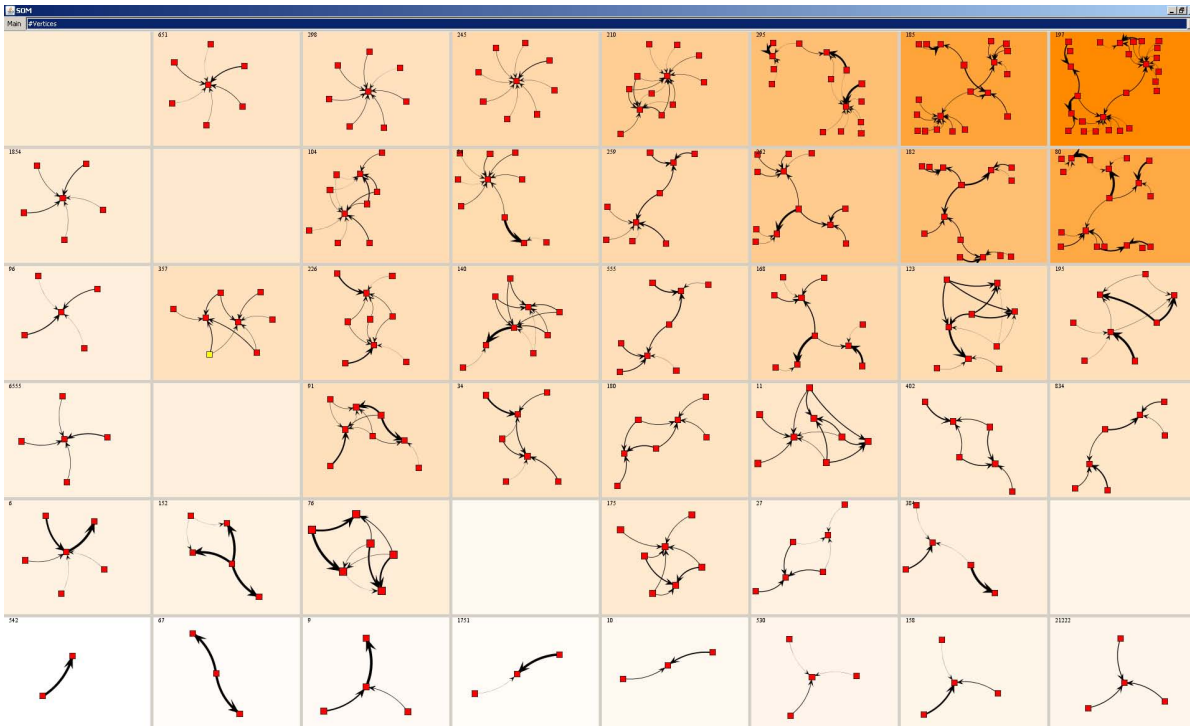


Figure 3.38.: Visualization of the SOM result with coloring of the cell background according to the values of a selected dimension of the SOM prototype vectors for the analysis of the SOM grid composition according to the selected feature.

parts of the cells on demand. Cells with a large number of members can also be used as an input for subsequent clustering. This allows for refinement of the clustering results as illustrated in Figure 3.39.

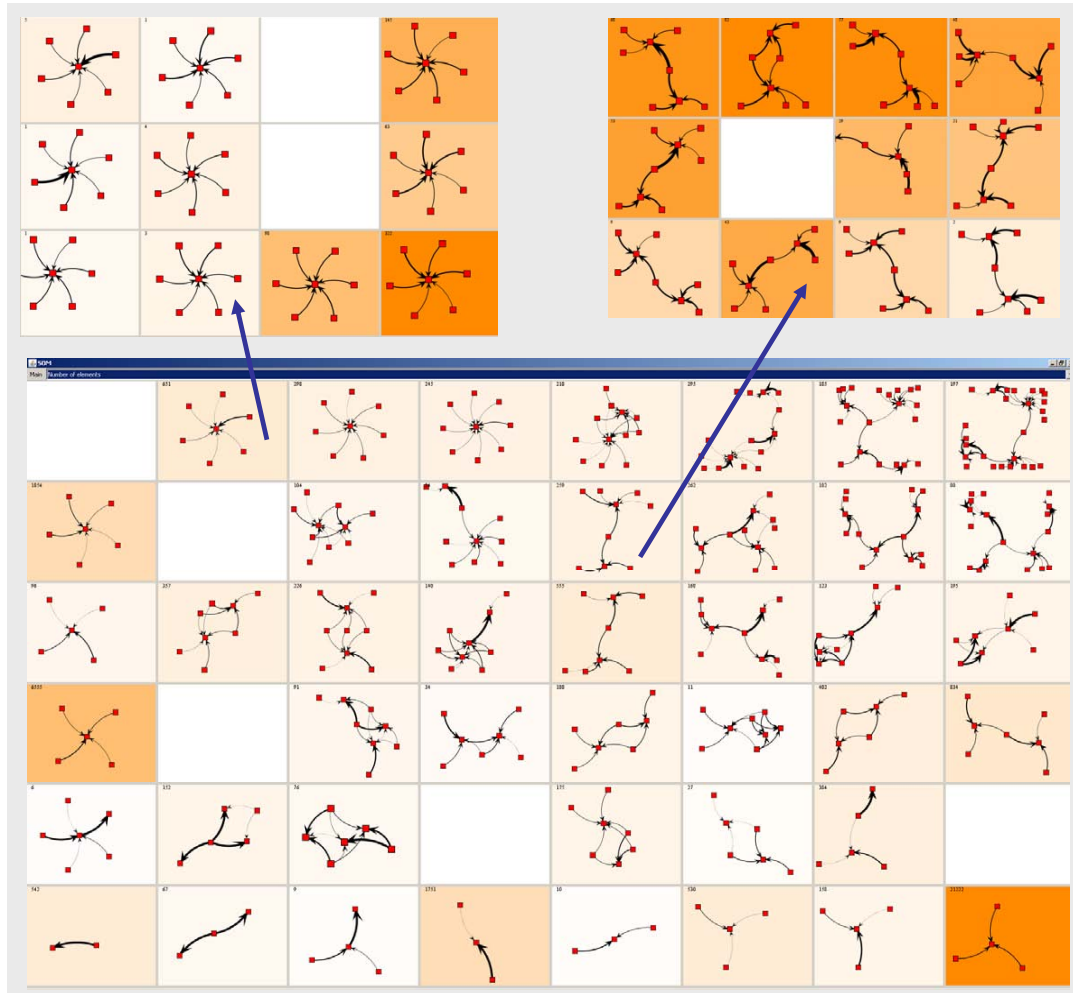


Figure 3.39.: Refinement of the SOM result using SOM clustering of selected SOM cells used for cells with large number of members and/or for further detailed inspection of the graph distribution in a cell. The figures on the top show the SOM grids created by clustering members of the cells in the initial SOM on the bottom of the picture.

3.6.6. Interactive Visualization of SOM Clustering Quality

In order to assess the quality of the clustering, several interactive views on the quality of results based on feature and distance distributions in cells and calculation and visualization of clustering quality measures are provided.

Please note that owing to the lack of direct representation of cell centers as graphs we rely on abstract graphical representations and algorithmic analysis.

The *distance distribution view* shows the distribution of distances between cell elements and the respective cell center (see Figure 3.40). In this view, we can see that the SOM clustering provides very good results, as most of the graphs assigned to a cluster are very close to the cluster center and only few outliers with larger distances appear. The outlier graphs are displayed in a member view, where the background color corresponds to the distance to the cluster center. For comparison, the cluster representative is shown as well (the top left graph in the view).

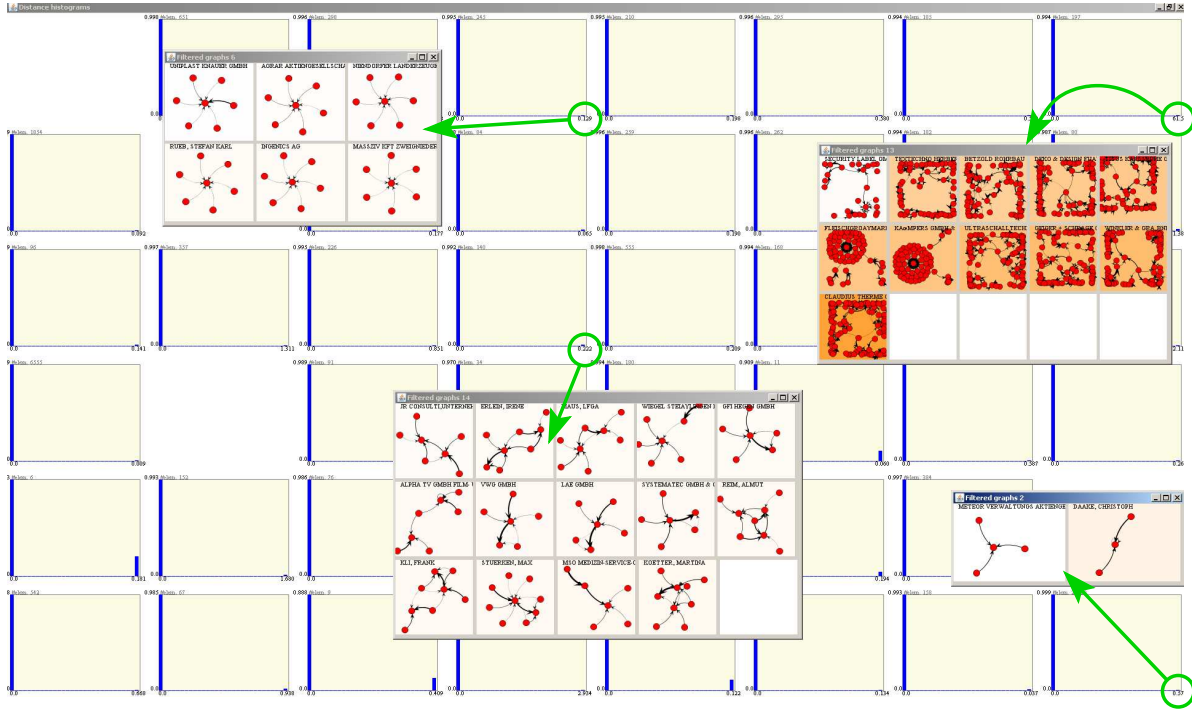


Figure 3.40.: The distribution of distances between cell members and cell center across the SOM grid are displayed in form of a histogram. Small distances indicate good clustering results therefore largely left skewed distributions close to zero are preferred (as shown in the figure). Larger distances at the end of the distribution indicate outliers. These cluster outliers together with cluster representatives are displayed in a separate window on demand. In this additional view, distance to the center is mapped to the background color.

Interactive Visualization of cell member distance and feature distributions The *display of feature distributions* for the members of the cells shows the range and frequency of graph features in a selected cluster. Together with the graph distance view (see Figure 3.40), it allows to spot outliers in the clusters and assess the overall quality of the clustering. On demand, an overview of the cluster members across the feature distribution or parts of the histogram can be displayed for detailed analysis of the cluster (see Figure 3.41).

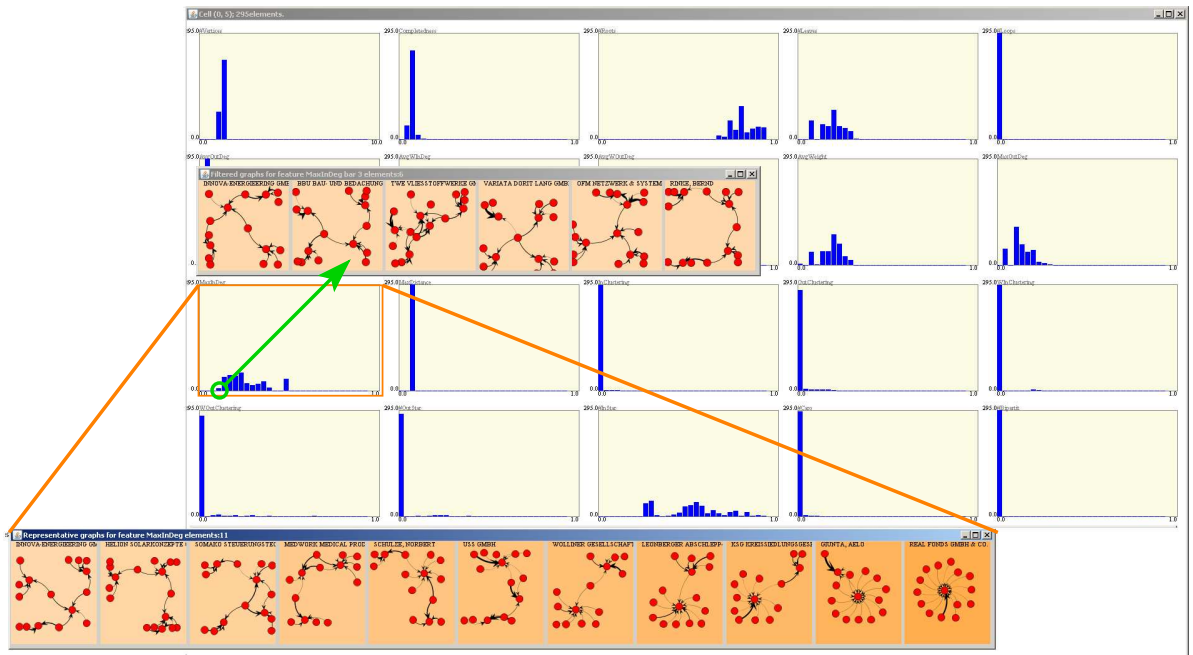


Figure 3.41.: Visual analysis of feature distributions for a particular SOM grid cell. The histograms of each feature in a cell are shown. Narrow distributions are expected for good clustering results. Wider distributions may indicate the need for inclusion of this feature in further clustering. On demand, an overview of the cell graphs, showing representative graphs from all parts of the selected feature distribution, is displayed in a pop-up window. This allows for inspection of the graph structure variance with respect to the selected feature.

The assessment of clustering quality using distance to cell centers is also supported by a parallel coordinates view showing values of the cell center features (displayed in orange) and cell member features using transparency. This view can be used together with the previous displays as an indication of performance of individual features. For more information on this type of display, we refer to the Section 4.6.6.

Visualization of algorithmic assessment of clustering quality In addition to the previous views, we provide the *assessment of clustering quality by various measures*. This allows for a quantitative assessment of the SOM quality and detailed inspection of the quality measure values across SOM cells. The currently used measures are based on the clustering quality literature (see Subsection 2.5.2.1). The overview of the measure values together with the distribution of a selected measure across the SOM grid are calculated and displayed in the SOM grid (see Figure 3.42 for an illustration).

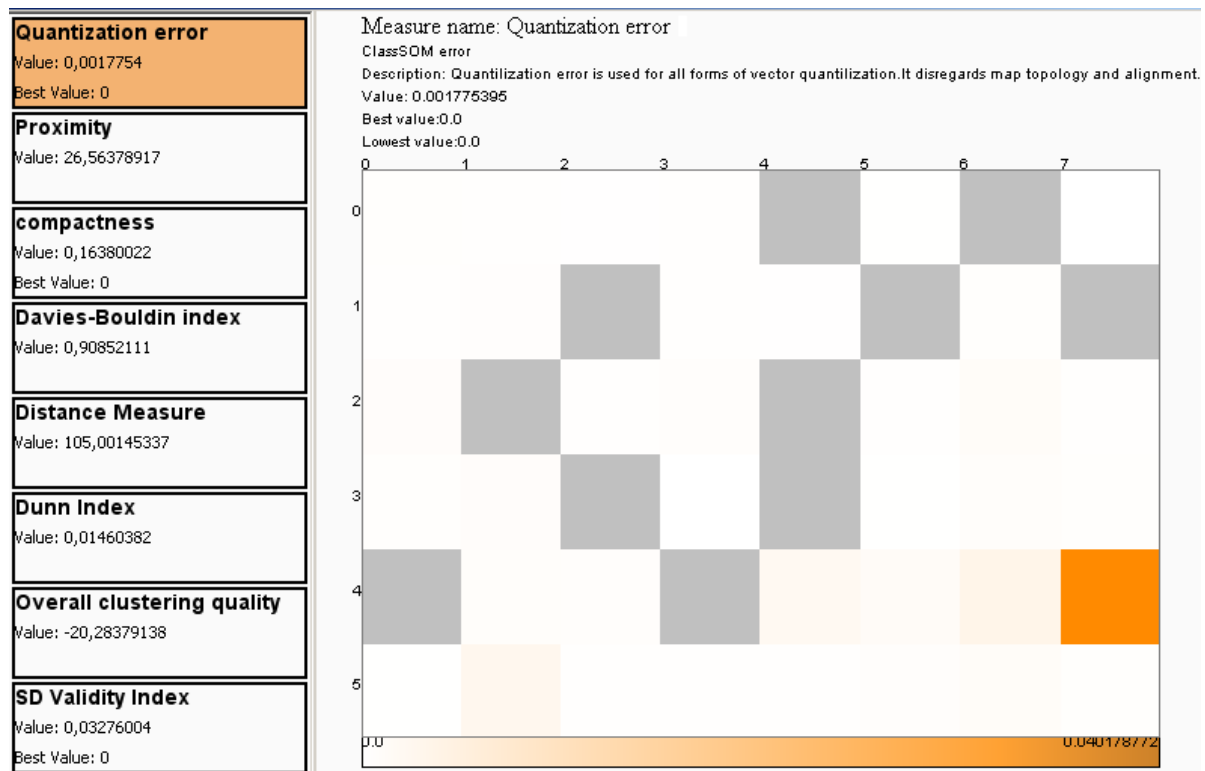


Figure 3.42.: Visualization of a selected SOM quality measure. The quantization error shows the average distance of cell members to the cell center. The measure values are displayed in the SOM grid using a heatmap. White color means low distances and orange color indicates cells with high distance. The grey cells are empty. This view shows good overall clustering quality with one cell with larger distance that can be inspected in more detail on demand.

3.7. Application

3.7.1. Introduction

In this section, we apply the previously introduced techniques to the visual analysis of shareholding networks. Shareholding relationships between companies in an economic system can be regarded as a weighted directed graph with nodes representing companies, and weighted, directed edges representing the “holds-shares-in” relationship between corporations. The edge weight represents the percentage of shares held in a company. Depending on the data source, further edge and node attributes may be provided (e.g., company size, company sector, number of employees, type of relationship). These attributes are used in the analysis on demand.

In a modern economy, large structures of cross-corporation, cross-border shareholding relationships exist. These form complex networks and dependencies between the players of all sectors of the economy. These shareholder networks in turn give rise to complex structures of investment and control between the involved entities. As stated in the report of the European Corporate Governance Institute (ECGI) [Eur07] – “*Control Enhancing Mechanisms are rather common in the sample of listed companies in European member states that are analyzed in [this] report. Of all the 464 European companies considered, 44% have one or more CEMs.*” The main CEMs are pyramids and cross-ownerships³. For example, according to the study, 15% of German, 45% Italian and 56% Swedish companies use pyramids.

The proper understanding of shareholder networks is essential for many important financial analysis decisions, e.g., for assessing the value of a firm [Eur07], for pricing corporate shares [Dam05] for analyzing the competitive position a corporation faces in the market, and thus whether to enter or leave a certain market. Depending on the application task, the focus of the analysis can be for example examination of a structure of a selected company, analysis of the company structures in the whole economy or their combination. The analysis may involve several steps (see also Figure 3.2).

On their web pages, in annual reports, the companies usually present their structure as simple lists (see Figure 3.48 for an illustration). The survey of the *economic papers* on shareholder structures (see Subsection 3.7.3.1) shows that the company networks are usually displayed as simple organigrams or as ownership matrices. In a few cases, freely available graph drawing software (e.g., Pajek or GraphViz) have been used. The pure presentation of data (as graph or organigram) shows the company structures as networks. However it does not provide analytic functionalities e.g., the identification of all shareholders of a selected company.

Although network and hierarchy visualization techniques, in general, are more widely applied in other areas of economic and financial analysis, only a few systems concentrate on the visualization of shareholder structures (e.g., [HK03, FAS04, the]). These systems however do not provide interactive features supporting the financial analysis tasks as they offer only visualization. Therefore currently available systems do not support the identified analytical tasks in an efficient and interactive way.

In the following sections, we show how shareholder analysis can be supported by our developed visual analysis tools. We concentrate thereby on the three types of tasks (as introduced in Section 3.1.1). We cover several user tasks and analytical processes. Firstly, we show an example of shareholder structure analysis on a selected individual shareholder network (Heraeus Holding) using interactive explorative methods. Secondly, the motif-based analysis is used for examination of shareholder substructures in the whole economy and identification

³Pyramid structures are built by a tree structure of company holdings. In the pyramid, the control is usually held by the shareholder on the top of the pyramid.

Cross-ownership structures are constructed by linking horizontally cross-holdings of shares that reinforce and entrench the power of central shareholders.

of interesting individual corporate structures which are analyzed in more detail in the following. Finally, we perform analysis of the types of shareholding networks in the whole German economy network.

3.7.2. Data

For the application, data on shareholding networks involving the German economy are examined. We consider the *Amadeus* [Bur] and *Hoppenstedt* [Hop] databases, which contain financial and ownership data on German and international corporations, which have subsidiaries in Germany. The shareholding relationships are provided as tables with lists of companies and public entities (and private persons in case of Amadeus) holding participation in a company and the relative amount of participation. Moreover, the databases include (at least to some extent) further data on the companies (e.g., number of employees, total assets, sector, etc.).

The provided data and the data structure in these two databases are suitable for a comprehensible visual analysis of the shareholding networks as considered in this thesis. We describe the structure of the two databases with regard to shareholding networks in the following showing the similarities and differences in their compositions.

The Amadeus database [Bur] used in the examples includes around 332,000 entities (companies, public bodies and private persons) and ca. 310,000 relationships between them. The whole shareholding graph consists of more than 40,000 weakly connected components. One is very large (with around 100,000 nodes (113,000 edges) and the rest are small (with up to 100 nodes each). Figure 3.43 shows a random sample of 120 shareholding graphs from the set of weakly connected components. The distribution of graph sizes and orders of the small components can be seen in the Figure 3.44 (please notice the logarithmic scale). The small components, in general, have multiple roots (on average 3.82) and a high proportion of leaf nodes with loops being an exception. The high number of root nodes (ultimate shareholders) in comparison to the Hoppenstedt database (see below) results from the inclusion of private persons in the data. The small components have a modest connectivity (on average 0.256). The large graph is very sparse and includes a relatively high proportion of roots.

The Hoppenstedt Database [Hop] from December 2008 contains 105,672 commercial companies, ranging from large corporates to small firms, as well as public sector entities. It covers around 124,000 shareholding relationships between them. It concentrates mainly on shareholdings between companies (not private persons). The companies included in the database are mainly German, but also encompass affiliates abroad. Similarly to the Amadeus Database the whole network contains one very large component with (ca. 101,000 companies connected via ca. 120,000) relationships. The other 59 components contain networks with up to 530 nodes. The distribution of selected network parameters for the connected components can be seen in the Figure 3.46 and one example of such network is presented in the Figure 3.45. The data shows that small components have in general one root node (corresponding to the ultimate shareholder) and very low connectivity (on average 0.07). Loops (edges from and to the same node meaning that the company corresponding to this node is holding shares in itself) occur only in the large component, which includes more than 270 root nodes and has also a very low connectivity.

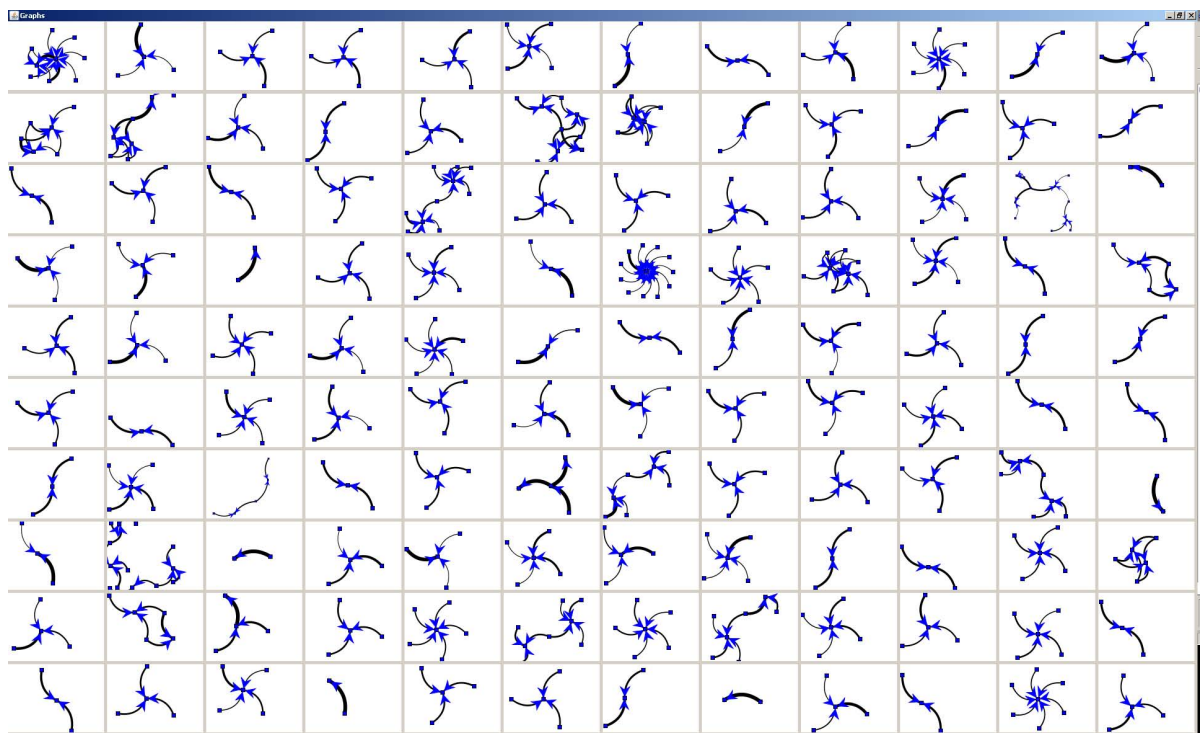


Figure 3.43.: Example of shareholding networks from Amadeus database showing a random sample of 120 shareholding graphs (connected components).

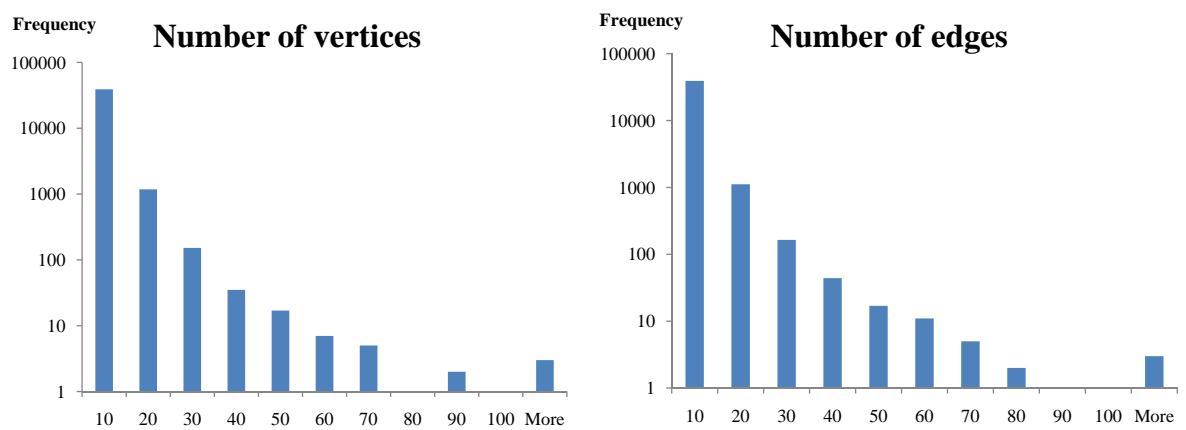


Figure 3.44.: Selected parameters of the connected components from Amadeus database showing the graph size and graph order distribution with strong decline in the frequencies. Note the logarithmic scale of the y-axis.

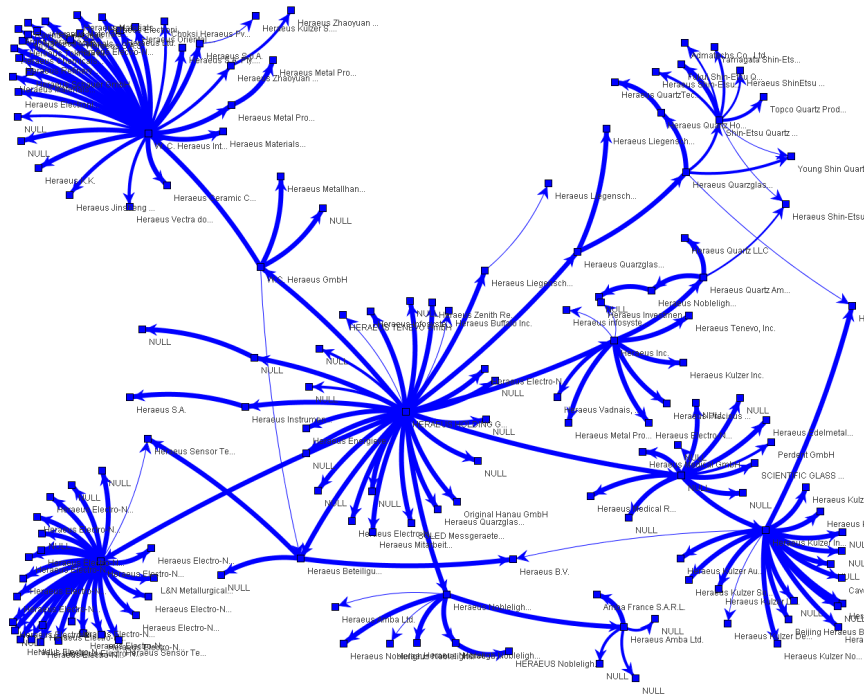


Figure 3.45.: Visualization of one selected corporate shareholding component from Hoppenstedt database.

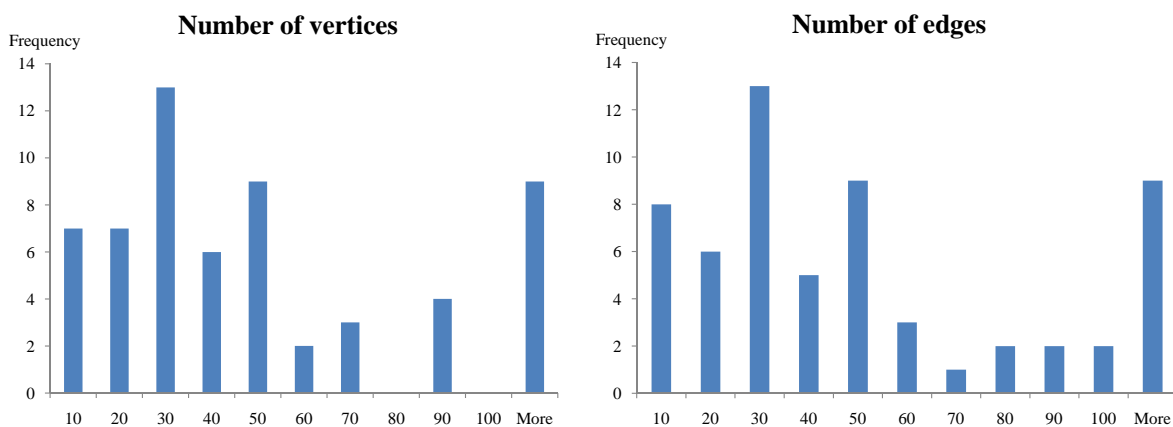


Figure 3.46.: Selected features of the shareholder networks in the Hoppenstedt database showing the distribution of graph size and order.

3.7.3. Visual Exploration of Shareholder Networks

In this section, we show how our system supports the analysis of individual shareholder networks. In this respect, we combine state-of-the-art visualization techniques with the graph algorithmic analysis and economic aspects described in the Subsection 3.7.3.1.

In the following, we first explain the financial theory of the shareholder analysis. Then we provide examples how our system enables visual identification of the ultimate shareholders and supports the visual analysis of integrated cash flow and control rights using real-world data. The tools presented in this section are going to be used in a real application by a well known German provider of shareholder data – Hoppenstedt [Hop].

3.7.3.1. Shareholder Analysis Theory

In order to identify relevant analytical tasks when exploring a corporate structure from an economic perspective, extensive interviews with financial analysts as well as a survey of the economic literature on corporate governance was performed. The results are used for implementation of relevant graph analytical functions for the visual analysis of shareholding networks. The economic studies [Wiw01, BM02, CGS03, LPLS99] mainly concentrate on the impact of the structures on the economic performance of the firms and their competitive behavior. The resulting main tasks are:

1. Identification of ownership structures,
2. Exploration of corporate control,
3. Identification of ultimate controlling shareholders.

In general, these tasks can be quite complex owing to various ownership structures used to control a company. As these structures tend to obscure the ownership and the control of the company, it is necessary to determine how to measure the degree of control of a company and the constellation of rights in order to identify the ultimate structure of control (see economic analytic tasks identified in Section 3.7.3.1). For example, “pyramid structures”⁴ are popular in Germany in the automobile sector (e.g., Volkswagen and MAN) and the utilities sector (e.g., RWE). A cross-shareholding between Allianz and Münchener Rückversicherung is a well-known example of CEMs: Allianz has a 9.4% stake in Münchener Rückversicherung which in turn has a 5% stake in Allianz.

An example showing ownership and control over cash-flows by company A over companies B to D in a pyramid structure is presented in Figure 3.47. In this example, we see the separation of control rights from cash-flow rights. Company A only holds 29% of cash-flow rights of company D and at the same time is the controller of this company through the control of company B. Company A is ultimate shareholder as it is not owned by any other shareholder.

Calculation of cash flow rights In order to answer the question “Who are the shareholders of a company and how much shares do they own” we need to calculate the so-called cash-flow rights (direct and indirect share holdings) of a company. In the article, we follow the approach of Chapelle et al. [CS02].⁵

Let a_{ij} represent the ratio of shares owned by company j to the total outstanding shares of company i . By definition $0 \leq a_{ij} \leq 1$ holds. The matrix $A = [a_{ij}]_{n \times n}$, therefore represents the direct share holdings of companies in the sample, while n is the number of companies in the system. Equation 3.1 represents the constraints of the shareholder network system⁶

⁴Pyramid structures are hierarchic shareholding structures often with several levels of shareholding relationship and several companies held by a company.

⁵We do not consider preferred voting shares in the calculation.

⁶In a closed system, where all shareholders are known the sum is equal to 1 for each company.

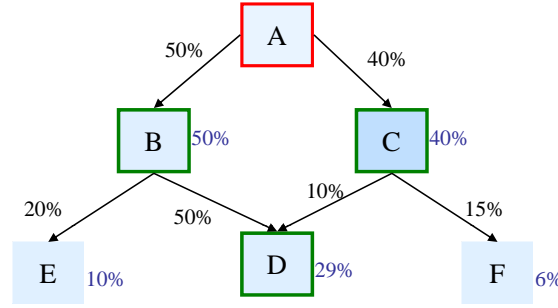


Figure 3.47.: Cash-flow rights, control and ultimate shareholder in a pyramid structure.

Vertices represent shareholders. Directed arrows represent shares ownership. Black weights of arrows represent direct shares ownership as a percentage of total shares. The blue figures represent cash-flow rights of the company A in the respective company. The root vertex highlighted with a red rectangle represents the ultimate shareholder corporation. The vertices highlighted with green rectangle represent corporations controlled by corporation A.

$$\sum_{j=1}^n a_{ij} \leq 1, i = 1, \dots, n. \quad (3.1)$$

The integrated control (cash flow rights) is represented by the matrix $V = [v_{ij}]_{n \times n}$ which is calculated as in Equation 3.2:

$$V = [\text{Diag}(I - A_1)](I - A)^{-1}A \quad (3.2)$$

where A_1 is defined as $A_1 = [a_j]$. Its elements a_j are defined in Equation 3.3.

$$a_j = \sum_{i=1}^n a_{ij}, j = 1, \dots, n. \quad (3.3)$$

Each element v_{ij} of matrix V represents the integrated (direct and indirect) shares of company j in company i . If $v_{ij} > 0$ then company j is a shareholder of company i and v_{ij} represents the amount of total holding of company i by company j .

Calculation of control rights The control rights of a company need to be calculated for answering the question “Who has a control of a company”, because this can influence the company’s performance and therefore its value. In this respect, Spanish and Italian shareholder structures tend to be particularly challenging and complex. For example, the Agnelli family maintains control of the Fiat group, although it owns only around 4% of the Fiat shares – through a cascade of two quoted companies, IFI and IFIL, in each of which it has a controlling interest. Using the presented application, the controlling relationships can be shown by highlighting the companies controlled.

The control over a company A by a company B is measured as a maximum control over the different control paths (i.e., all shareholder relationship paths between companies A and B). The control in each shareholding path is defined as a minimum of direct controls in the path. The direct control is calculated as follows: if a direct share in a company is more than the control threshold (e.g., 30%) we set the direct control to 100%, otherwise we set

it to 0%. The threshold value is not universal as it depends on the definition of company control. For example, in Germany it is set to 30%, as stated by the German Financial Market Supervisory Authority [Bun08]. In other countries it may be 50% or another threshold value.

3.7.3.2. Results

In the following, we illustrate the visual exploration capabilities of our system on a shareholder network of the German company Heraeus Holding GmbH taken from the Hoppenstedt Database. Heraeus is a globally active precious metals and technology group, headquartered in Hanau near Frankfurt am Main. The focus of its activities is on special and precious metals, sensors, dental products and bio-materials, quartz glass, and specialty lighting sources www.heraeus.de. It is present with subsidiaries in 35 countries worldwide.

Heraeus Holding GmbH presents its corporate structure as tabular list of companies. The list is divided into the six business segments and four geographical areas (see Figure 3.48). While this illustration gives an overview of the company activities, it does not provide many important aspects that a financial analyst would require in order to assess the company. More specifically, the tabular form does not include any mention of relationships between the firms listed. From this table it can not be seen whether the German activities of Heraeus are controlled through the foreign holding companies or directly by the top holding company, for instance, and whether their are holding sub-structures.

Figure 3.49 shows the visualization of the corporate structure of Heraeus Holding GmbH with 160 nodes and 165 connections. In this figure, the six main business areas of the firm are clearly identifiable in the different star-shaped relationships. The figure also illustrates the intricacies in the holding pattern of the firm, which were not visible in the tabular presentation (see Figure 3.48). For instance, it is clearly presented that the activities in the United States of America form a separate star-shaped holding structure, while this is subsumed in the table. Such information is important in order to understand the functioning of the corporation, which is a key ingredient into the assessment of default risk or valuation of the company for investors, regulators and banks.

1. Identification of corporate shareholdings We first concentrate on the identification of companies held by a selected company and the identification of all shareholders of a company. As an example we use W.C. Heraeus as the origins of the corporation go back to this company. It still plays a central role in the group as holder of the metallurgical business activities providing the largest share of revenues. Figure 3.50 left shows all (also indirect) companies held by the company W.C. Heraeus. The holdings of this company form a pyramid with many substructures and at several levels. The only shareholder of the company W.C. Heraeus (see Figure 3.50 right) is the group holding company Heraeus Holding GmbH.

Additionally, it is possible to identify companies having direct shares in many companies (resp. owned by many entities). Figure 3.51 shows examples of companies holding at least 30 entities (left) and at least 20 companies (right). We can see that Heraeus Holding (see also following paragraph) owns shares in 30 companies (being maximum number in this network). Setting up the threshold (minimum number of companies held) to 20 companies reveals another two companies holding shares in many companies, namely W.C. Heraeus and Heraeus Nite. The first is the center of the metallurgical activities, while the second is the central unit of the electronic sensor business area – the two largest segments of the corporation. Smaller business areas such as quartz glass and specialty lighting sources are not picked-up by this procedure. In summary, while the complexity of a network may not be representative of the revenue/profit situation, it does provide a correct indication as where the corporations places emphasis with regard to its central activities.

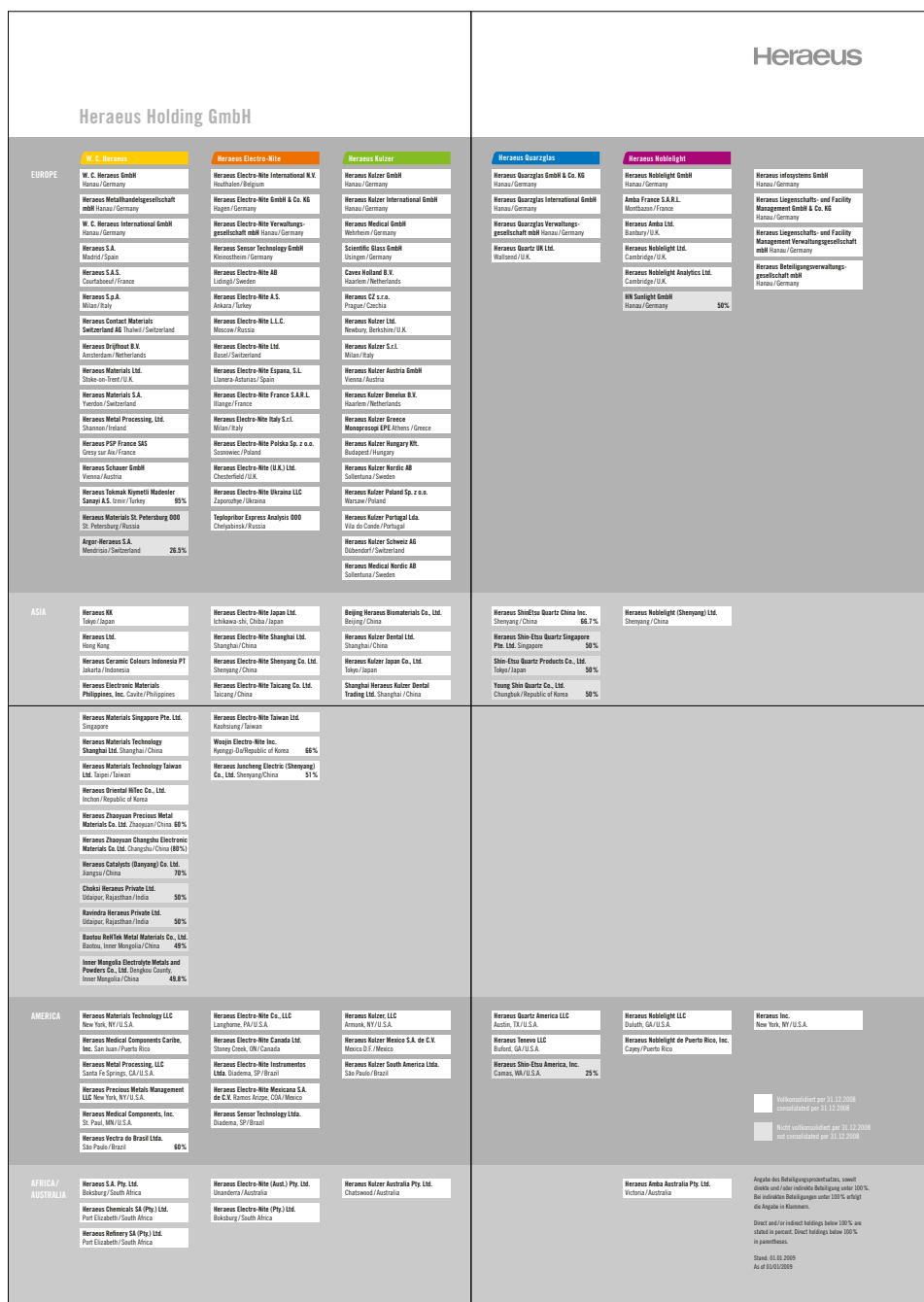


Figure 3.48.: The corporate structure of Heraeus Holding as presented on its website www.heraeus.de. This type of presentation does not reveal the holding structure of the company.

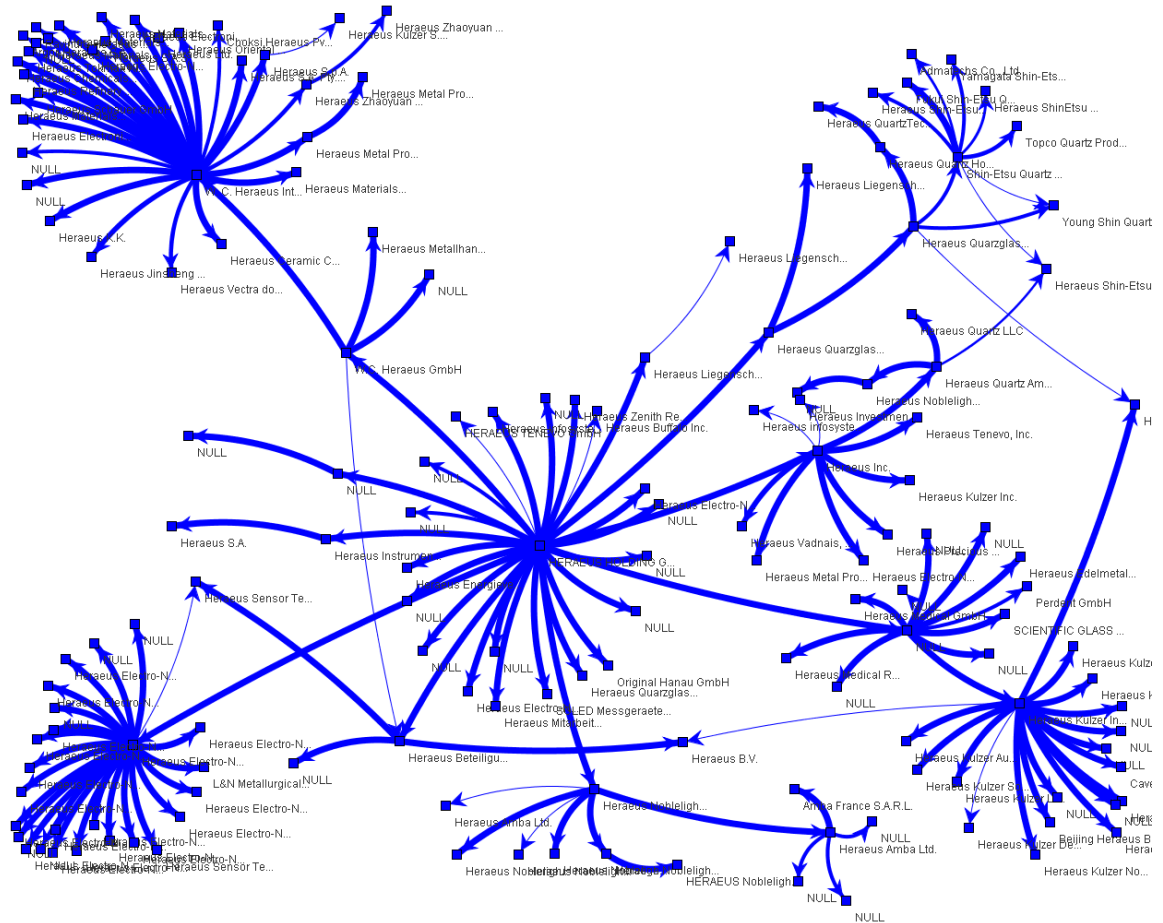


Figure 3.49.: Visualization of the shareholder network of the company Heraeus Holding GmbH showing the six main business areas composed in the different star-shaped relationships. The companies are nodes of the graph. The edge direction (edge arrow) shows the “holds-shares-in” relationship. The edge thickness represents the ownership share. The labels show company names, the “NULL” values result from missing values in the database.

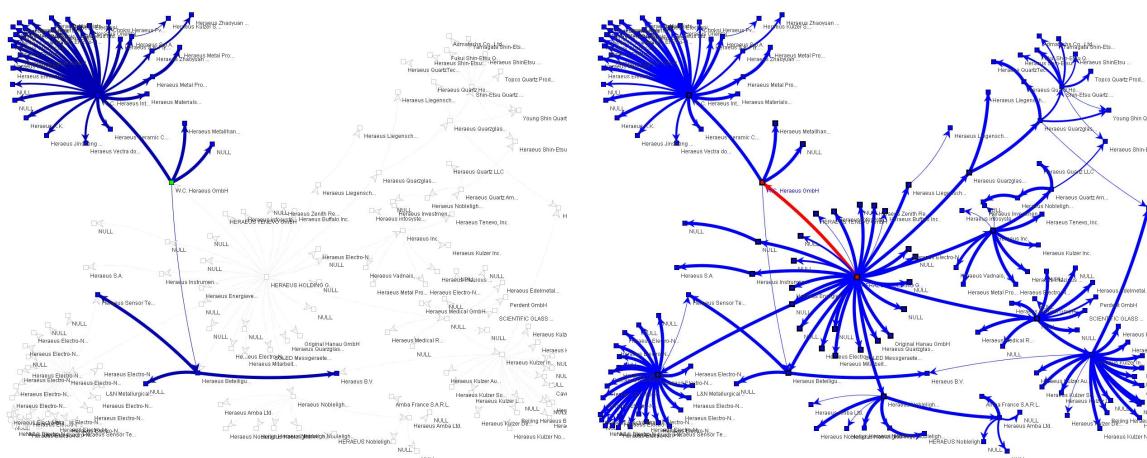


Figure 3.50.: Visual analysis of integrated shares. Left: Companies held by W.C. Heraeus GmbH. The blue color highlights the companies and relationships of interest. Right: Companies owning shares in W.C. Heraeus GmbH. Red color highlights the holding company (Heraeus Holding GmbH.) and the holding relationship.

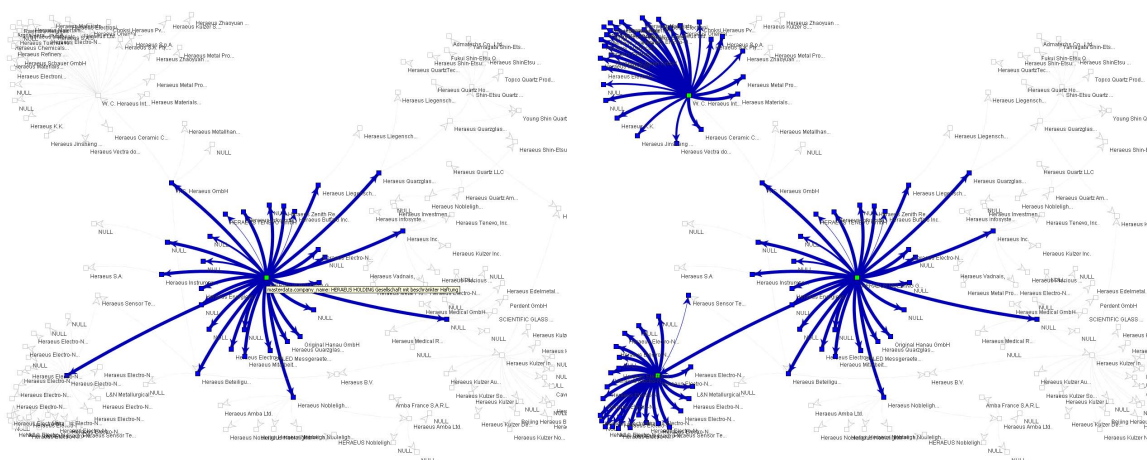


Figure 3.51.: Important companies holding shares in many companies. Left: Using the threshold of at least 30 companies held, one holding company with their children results. Right: Using the threshold of at least 20 companies held, three company sub-structures appear. The holding companies, holding relationships and held companies are highlighted in blue.

2. Company control and ultimate shareholders The question “Who is at the top of the pyramid?” can be answered by highlighting the ultimate shareholders in the visualized network, i.e., entities that are not owned by other entities and companies controlled by them. Figure 7a) shows the ultimate shareholders in the network. In this example, Heraeus Holding is at the top of the pyramid. Figure 3.52 shows the highlighting of companies controlled by a selected company. In our example, we use the controlling threshold of 50%. Please note that there is a possibility of interactively adjusting the threshold for the company control, as the definition is not harmonized across states. From the figure, it can be seen that the company Heraeus Holding controls almost all companies in the network, although it does not hold stakes in them directly. For example, as an exception it does not control Heraeus Shin-Etsu America Inc., which is part of a series of Joint Venture companies in the US and Asia of the Heraeus quartz glass division with a partner, Shin Etsu Chemicals.

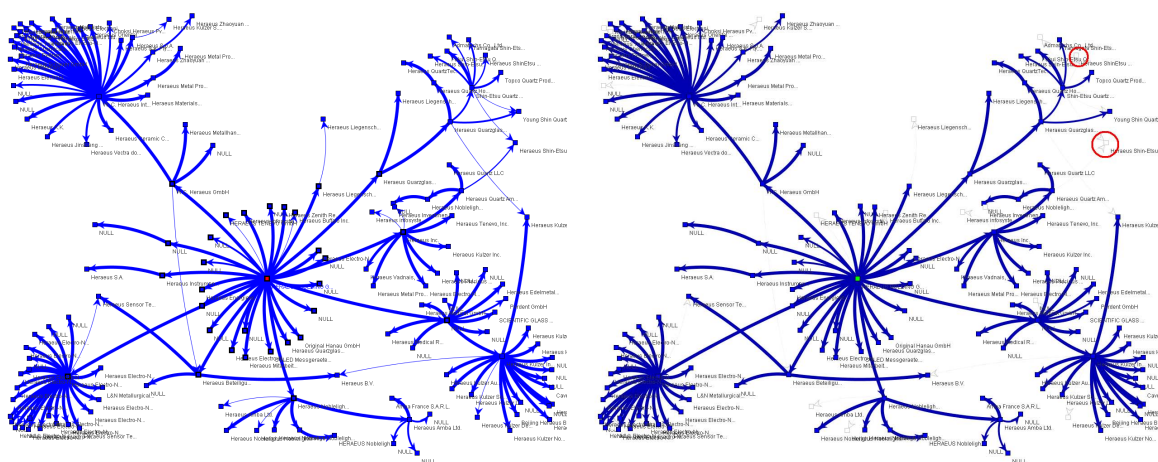


Figure 3.52.: Identification of ultimate shareholders and controlling rights. Left: The ultimate shareholder (company Heraeus Holding) of the Heraeus corporation highlighted in red color. Right: Companies controlled by the ultimate shareholder at at least 50% level are highlighted in red. Selected companies not controlled by the ultimate shareholder are emphasized within red circles.

3. Analysis of relationships between two companies The interactive visualization system offers the possibility to analyze the type of relationship linking two selected companies. These relationships are identified and then displayed in the graph. For example, the shareholders common to the two companies or the companies jointly held by both companies can be shown. If the companies are related, the relevant relationships are highlighted (see Figure 3.53). The user is informed about the type of relationship and the two companies are highlighted and the companies connecting them as well.

The example below shows the connection between two business areas in Heraeus corporation represented by the firms W.C. Heraeus GmbH and Young Shin Quarz Co. Ltd. (see Figure 3.53 right). It shows that the relationship is given by a common parent company (Heraeus Holding) which is the main node connecting the business areas in the corporation. This is confirmed by the fact that there is no cross connection without involvement of the ultimate shareholder between the two companies. The connection to the Young Shon Quarz company at the end of the holding chain is twofold – from the company Heraeus Quarzglass International GmbH. directly and indirectly via Shin Etzu Quarz Products Co Ltd..

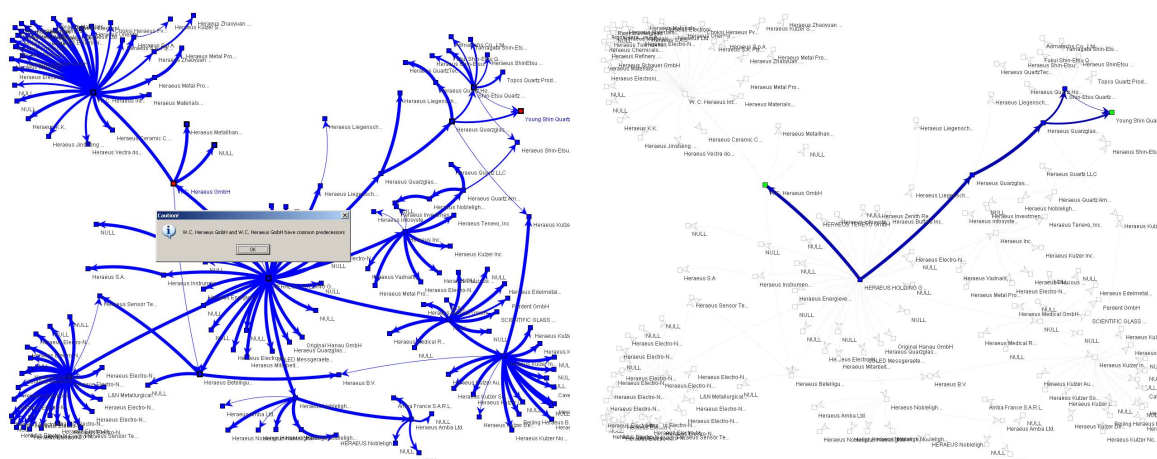


Figure 3.53.: Visual analysis of the relationship between two selected companies. In this example, W. C. Heraeus GmbH and Young Shin Quarz Co. Ltd. are examined for their relationship. Left: The whole network with the two selected companies highlighted in red color and type of identified relationship displayed. Right: The identified paths connecting the two companies of interest are highlighted in blue color.

3.7.4. Visual Analysis of Shareholder Networks based on Motifs

In this section, we show how the system is capable of revealing interesting shareholding substructures (motifs) in the networks considered. Several motifs are highly relevant for the analysis of shareholding networks (see below for detailed explanation). For example, some kinds of motifs can be used as control-enhancing mechanisms or to identify important entities in the network. With regard to motif analysis, the tasks introduced in Section 3.1.1 can be applied to shareholding analysis as follows:

1. Analysis of shareholder networks for specific motifs relevant for financial analysis.
2. “What-if-analysis” showing impacts of changes in shareholding relationships on substructures (motifs), e.g., when buying/selling shares or in case of company default/creation.
3. Analysis whether and what types of connections exist between specific shareholding motifs (i.e., specific/interesting shareholding substructures).

In order to present the application of the techniques developed for such tasks, we first explain how motifs defined in Section 3.5.2 apply to analysis of shareholding networks and then show results of motif analysis on the whole Hoppenstedt database and on an example corporate network (British-American Tobacco) from the same data set.

3.7.4.1. Motifs in Shareholder Networks

We now describe the motifs introduced in Section 3.5.2 with respect to shareholder analysis. The motifs used in this section are presented in Figure 3.54.

- *Feed-forward motif* is a simple structure which is usually used to perform control over a company indirectly via holding shares in a third company. The sum of shares held in a company directly and also via a third

company allows for higher voting power. When looking at edge weights in the motifs, such structures usually have weights below 50% for each edge, but sum up to more than 50 percent share in the final company thereby gaining control over this company.

- *Caro motif* This motif is the extension of simple voting power enhancing mechanisms (see feed-forward motif) involving four companies. The weighted case is an analogue to the example above.
- *Feedback motif* is a typical simple case of self holding where each company in the motif is held ultimately by itself via a third company. We refer to it as *indirect cross-holding*. In a parameterized case, we can pose constraints on maximum or minimum percentage of shares self-held by the companies (all or at least one of them).
- *(Extended) double cross motif* represents a case when two (or more) entities hold shares in the same companies, while often not being able to control the companies on their own. In case of a coalition the owning companies would however be able to control the companies. We can pose constraints on the minimum sum of shares held by both entities in one of the companies or in both companies (e.g., more than 50%).
- *Out-Star motif*: Out-Stars are composed of entities who own shares in a large number of companies. We can define motifs having a minimum number of companies held (we use four as default).
- *In-Star motif*: In-Stars are composed of companies who are owned by a large number of entities. Such companies often have a lot of small (minority) shareholders with little individual influence on them and a couple of larger shareholders who, although they are not majority shareholders control the company.
- *Reciprocity motif*: shows the case if two companies hold shares of each other. This motif is also called *direct cross-holding* when talking about shareholder networks.

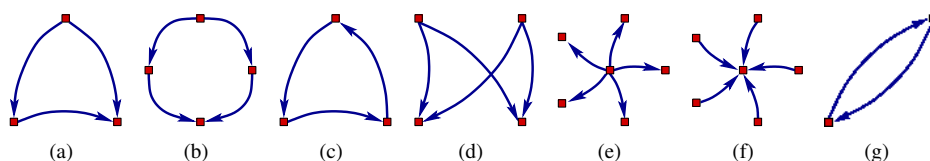


Figure 3.54.: Selected graph motifs. a) Feed-forward, b) Caro, c) Feedback (indirect cross holding), d) Double cross, e) Out-star, f) In-star and g) Reciprocity (direct cross holding).

3.7.4.2. Results for the Whole Economy

Motif analysis can be performed for getting overview of the types of structures occurring in the whole economy. The number and type of structures can be important. Moreover, it is interesting which companies are included in specific types of structures, whether these motifs are formed within one corporate structure, within one sector or across corporations/sectors. In the following we show results of such analysis for the whole Hoppenstedt Database containing more than 105,000 companies as an approximation to the analysis of the German shareholding network. The visualization of the whole German network may not reveal such structures, therefore we use the motif search for finding interesting patterns in the data. We describe results for selected motif types.

Firstly, the analysts may be interested in motifs forming *direct and/or indirect cross-holdings* (i.e., reciprocity and feedback motifs). In these types, the companies hold shares in themselves via second, and third company respectively. The search for the reciprocity motif (direct cross-holding) shows that there are 88 such cross holdings of including 2 to 5 companies each (see Figure 3.55 top for an overview and Figure 3.55 (bottom) for a

selection of companies with larger motifs). In total 194 companies are involved in direct-cross holding relationships. A similar analysis for feedback motifs shows 22 motifs including 3 to 5 companies each (see Figure 3.56) with 71 companies in total. Interesting insights result from these outcomes. In general, according to the names of companies, cross-holdings are formed within corporations and within sectors. An interesting example is the Allianz SE, which forms both direct and indirect cross holdings (reciprocity and feedback motifs) with other companies. It is involved in several such structures within its corporation and also within the industry (together with Muenchener Rueckversicherung und Deutsche Bank). The financial industry is very often present in the cross-holdings (see several banks and insurances (“Versicherungen”) in both views). However there are examples from other industries such as automobile (Nissan and Renault), and energy (EON and Energieversorgung, or Energie Suedwest).

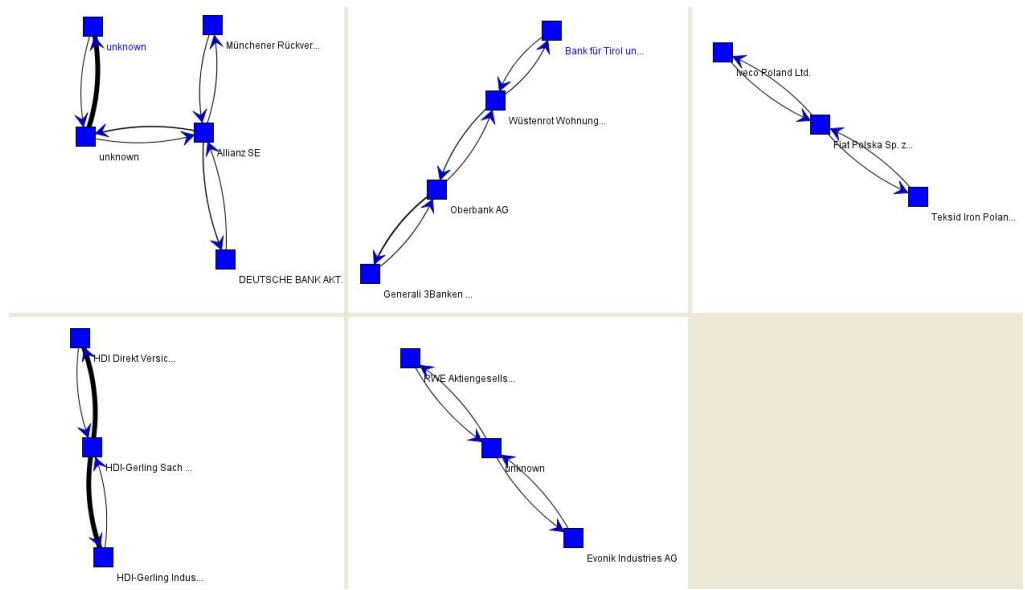
The analysis of *feed-forward* and *caro motifs* shows where one company holds shares in a second company via third (and fourth resp.) company. Such a structure is usually used in order to strengthen the control over the subsequent companies and an evaluation shows that this type of shareholding structure is very popular. Disconnected feed-forward motifs occur in 683 times (including 12,090 companies and 20,534 (overlapping) motifs) and disconnected caro motifs 220 times (including 4,745 companies and 3,437 (overlapping) 9,926 caro motifs). This is evidence for the high popularity for this type of control sub-structure. The largest connected feed-forward motif includes 5,955 companies, while the largest caro motif has 2,894 companies. The largest connected motifs structures relate to groups of companies connected to several large institutions, such as for example, IVG Immobilien AG (a real estate company), UBS (a bank), Allianz and Muenchener Rueckversicherung (both insurances). Feed-forward motifs are also very popular, with large formations identified inter alia at British-American Tobacco (cigarettes), Nestle (food), Phillips (electronics), or Unilever (diverse products). The corporations with large caro-shaped motif include for example Procter and Gamble (diverse), EDEKA (food), ENI (Petroleum, Gas), British-American Tobacco (cigarettes).

Double cross motif, which allows at least two companies to form a coalition in order to control at least two other companies, is found 344 times in the database. It creates 63,757 possible two-holding – two-controlled company combinations. The largest group of cross-held companies includes 7,347 entities. The largest corporations using this type of structure are Procter & Gamble and British-American Tobacco (discussed in the next section), while Nestle (food), Bestfoods Knorr (food) and Philips (electronics) also stand out.

Two corporate structures stand out as particularly *interesting* from the previous analysis. They are complex and exhibit a large number of motifs. The companies are the *Unilever group* (see Figure 3.57) and *British-American Tobacco* (BAT). We first discuss Unilever and then in the following section concentrate on BAT.

The Unilever group can be used to illustrate almost all previously mentioned motifs, more specifically it incorporates the second largest caro motif (see Figure 3.58 (left)). The figure shows that Unilever plc. and Unilever N.V. hold the participation in 127 companies around the world being the two parent companies of the Unilever Group. They are holding and service companies of the group, while the business activity of Unilever is carried out by the subsidiaries. Statutorily, shares in the subsidiary companies may ultimately be held wholly by either of the entities on its own or by the two companies in varying proportions. In fact, Figure 3.58 (right) illustrates that the companies at the end of the caro motif are held by both the Unilever companies, and thereby form a large double-cross holding structure. At the same time, a predominance of Unilever N.V. is documented by the fact that it has direct holding in over 670 companies (with 520 uniquely held entities).

Unilever plc. and Unilever N.V. have separate legal identities and have different shareholder constituencies. Shareholders cannot convert or exchange the shares of one company for shares of the other. NV is listed in Amsterdam and New York. PLC is listed in London and New York. However, in order to ensure unity of governance and management and thereby allow Unilever to operate as a single business entity, a special control



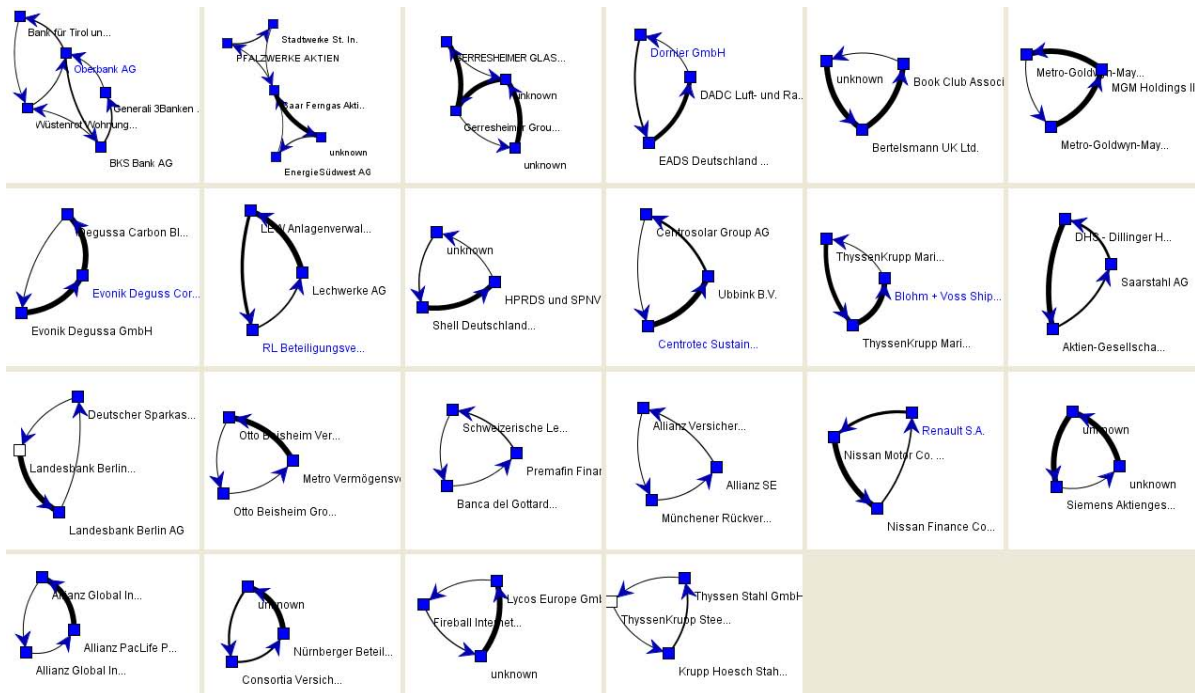


Figure 3.56.: Companies forming feedback motifs (indirect cross holdings) in German economy. Three larger structures in the left corner and several three company motifs are identified. Interestingly, the comparison of the identified direct and indirect cross holdings (see also Figure 3.55) reveal both types of structures within Allianz and Muenchener Rueckversicherung group.

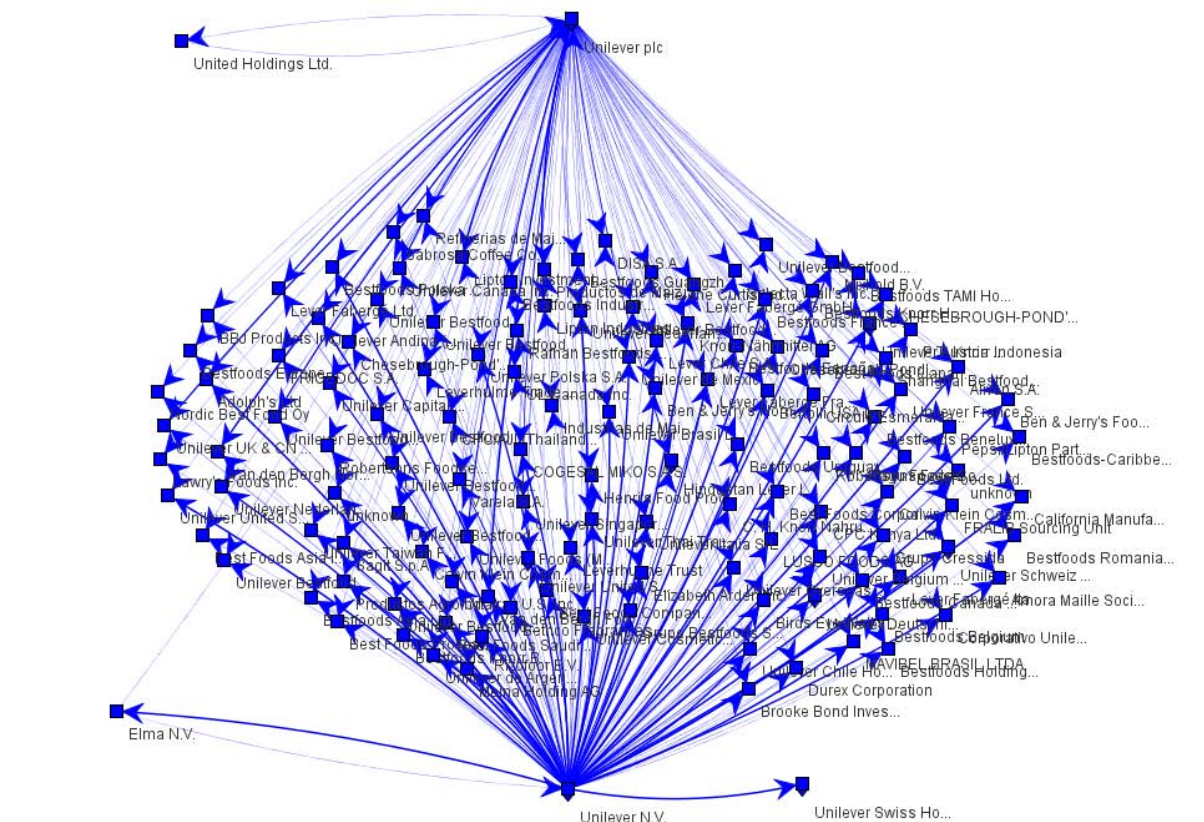


Figure 3.57.: Visualization of a part of the Unilever group identified by motif detection tools using several motif types. It shows four important companies (United Holdings Ltd., Unilever plc., Elma N.V. and Unilever N.V.) which are strongly involved in multiple corporate substructures.

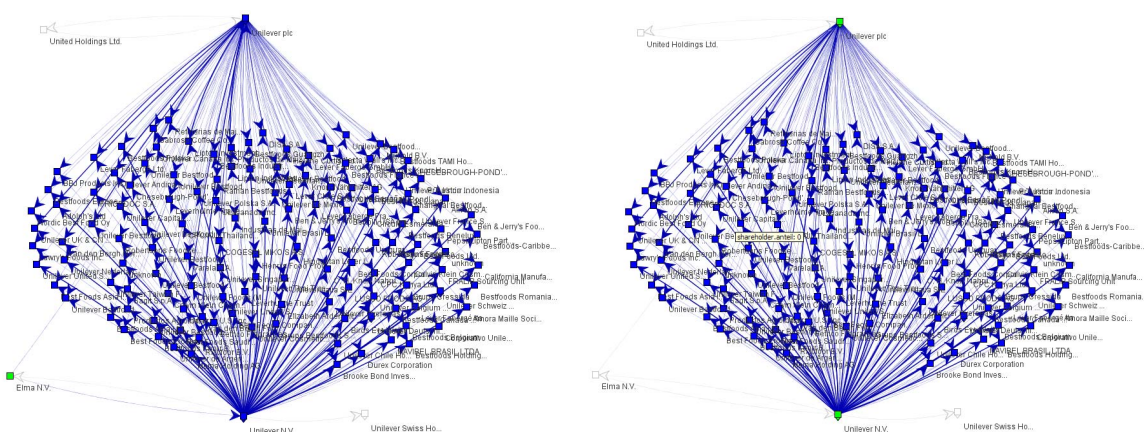


Figure 3.58.: Motif analysis of the Unilever group. Left: Caro motifs found headed by Elma N.V. highlighted in green color. Right: The found double cross motifs headed by Unilever plc. and Unilever N.V. highlighted in green.

structure between Unilever plc. and Unilever N.V. has been implemented by a legal agreement [Uni09]. This “Equalisation Agreement” regulates the mutual rights of the two sets of shareholders, including dividends. It sets out the co-operation in all areas, ensures that all group companies act accordingly and constrains the directors in the two companies to be identical.

Unilever plc. and Unilever N.V. are controlled on the basis of special rights by N.V. Elma and United Holdings Limited, which are joint subsidiaries of NV and PLC [Uni09]. For instance, the directors cannot be changed without the permission, of the holders of the special shares. The special shares may only be held or transferred to one or more other holders of such shares, which are N.V. Elma and United Holdings Limited. The implementation of this complex control structure is visible in Figure 3.59, which embodies two reciprocity loops and one feed-forward motif between the four companies involved. It also highlights the direct and indirect relationship (feed forward) between Unilever N.V. and Unilever plc. (also via Elma N.V.).

3.7.4.3. Results for a Selected Corporate Network

In the following, we use an example of the company British-American Tobacco (BAT) plc. indicated as being of interest in the previous analysis. BAT is a leading international tobacco company, which has a diverse portfolio of brands. The four “Global Drive Brands” are Dunhill, Pall Mall, Kent and Lucky Strike, but also Benson & Hedges and Rothmans. The graph for BAT has 532 vertices and 741 edges (see Figure 3.60). It includes 1 root node (i.e., a node without incoming edges or ultimate shareholders) and 443 leaves (i.e., nodes without outgoing edges or companies at the bottom of the shareholding pyramid).

Analysis of Shareholding Networks for Important Motifs In the selected example, the network shows a structure with many incoming and outgoing stars (i.e., companies being held by many entities or entities holding shares in many companies). The root node of the structure is British American Tobacco plc., located in London, UK. This unit controls mainly the firms owning the brands directly and British American Tobacco Finance plc., the financing hub of the group. The ownership over the other parts of the corporation are managed through two major “controlling” companies, existing in parallel: BAT Investments Ltd. and British American Tobacco

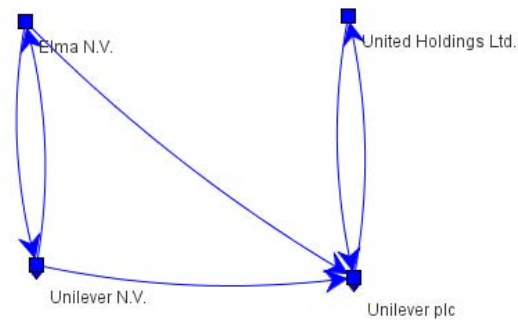


Figure 3.59.: Selected part of Unilever group showing relationships between the main companies Unilever N.V., Unilever plc., Elma N.V. and United Holding ltd..

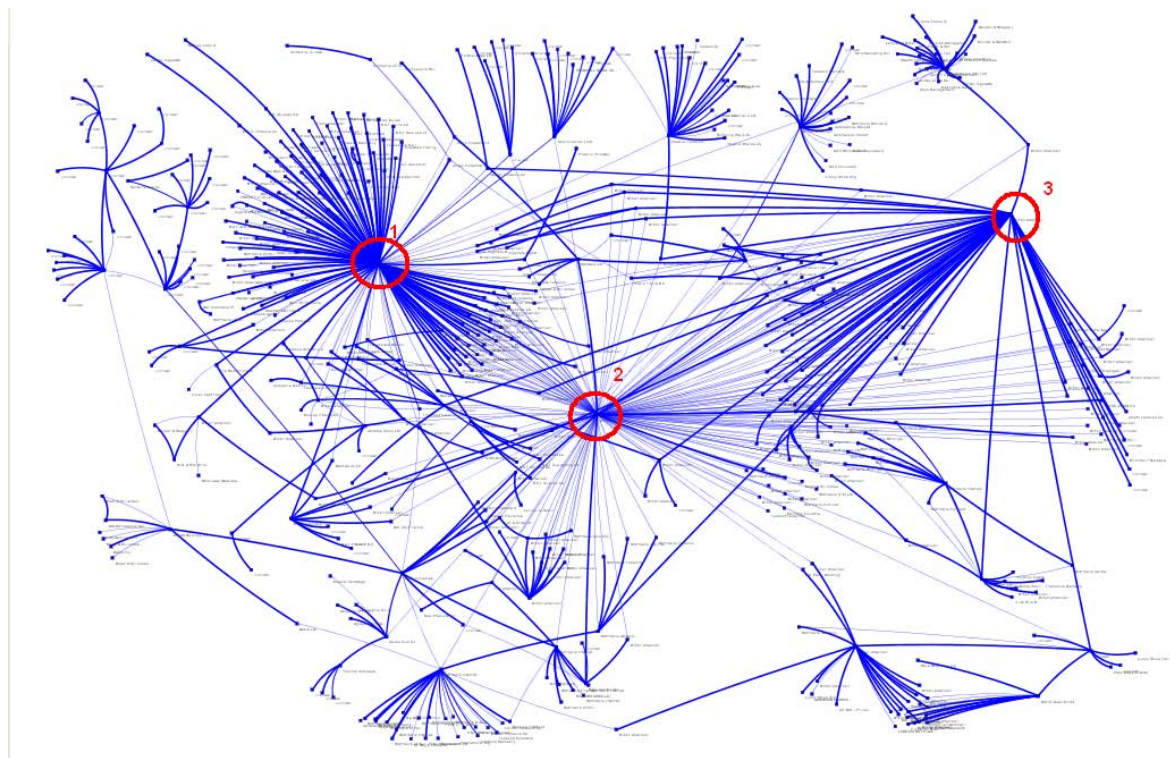


Figure 3.60.: The corporate structure of British American Tobacco plc. showing the complex interrelations within the corporate network with main companies BAT Investments ltd. (1), BAT plc. (2), and BAT International (Holdings) B.V. (3) highlighted in red circles. These companies are identified by the motif analysis.

International Holdings (UK). The first holds participations in 188 firms across the world – it is the largest outgoing star – while the second firm has 57 participations. This is visualized by Figure 3.61, which shows a feed-forward motif operating across BAT International Finance plc., toward the BAT Investments Ltd. A second chain runs by BAT International Finance plc. toward the BAT International Holding B.V., Netherlands.

The Figure 3.62, illustrates that the BAT group is characterized by a large number of *feed-forward* motifs which result from the fact that the subsidiaries are mainly held through one of the two holding companies mentioned above, but at the same time linked to the root node by a small direct participation. In addition, the figure also shows that this motif is found in a number of substructures, for instance centered around Rothmans plc., the sales and distribution business of British American Tobacco in the UK (red box) and for the activities of British American Tobacco in Australasia (red circle).

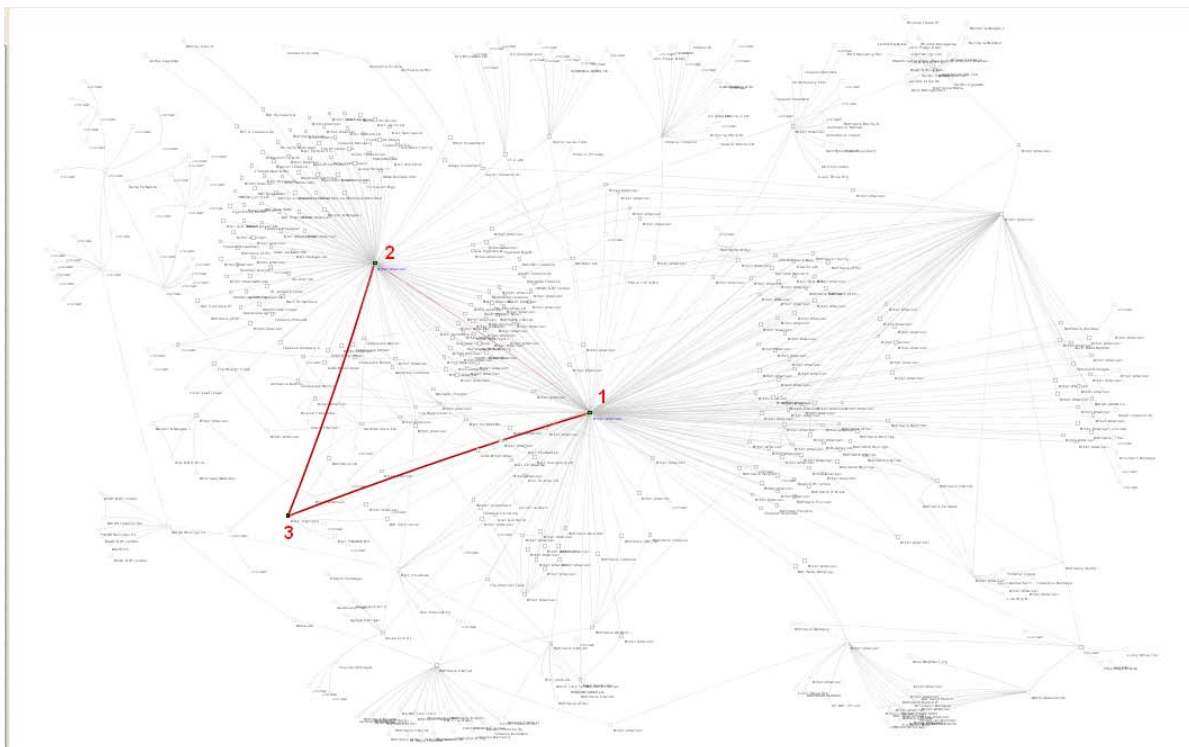


Figure 3.61.: The financial structure between BAT plc. (1) and BAT Investments (2) in the context of the whole BAT network. BAT plc. is holding stakes in BAT Investments directly and via a third company BAT International Finance (3) thereby forming a feed-forward motif.

A further frequent motif found in the ownership structure of British American Tobacco plc. is the *caro*. This motif allows control of a company by holding its shares via two other companies. Figure 3.63 shows that many edges not highlighted by the feed-forward motifs are elements in the caro. The caro motif identifies a number of subsidiaries, which are controlled by both main controlling companies directly. A typical example for such a motif is the operation of BAT in Nigeria, which runs across several different arms of the group (see Figure 3.64). This feature may have resulted from the fact that operations in Nigeria have been a greenfield market entry, requiring a bundling of competencies.

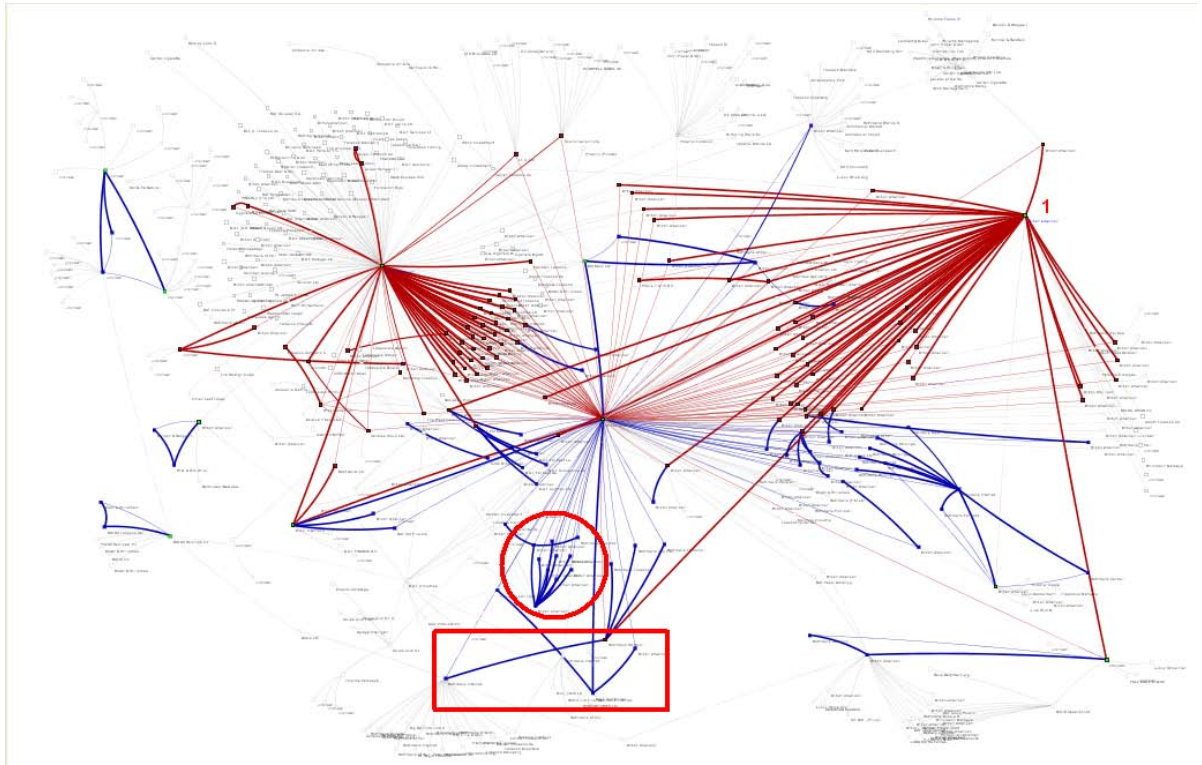


Figure 3.62.: Feed-forward motifs in the structure of British American Tobacco plc. (blue and red edges). Highlighted in red are motifs in which British American Tobacco International Holding B.V. (1) holds participations. The companies in the red circle identify the activities of British American Tobacco in Australasia. The red rectangle highlights the sales and distribution business of British American Tobacco in the UK. Both activities are found within the feed-forward motif analysis.

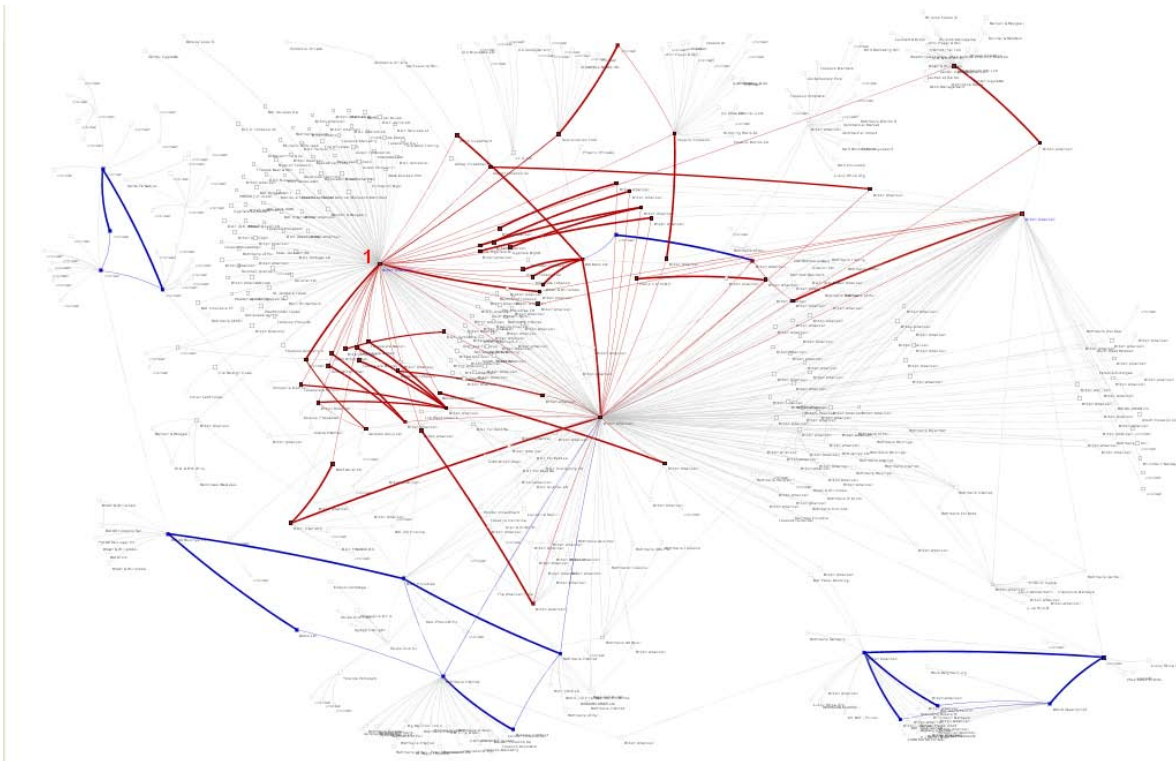


Figure 3.63.: Caro motifs in the structure of British American Tobacco plc. (blue) with red-highlighted motifs in which the company BAT Investments (1) is involved.

3. Visual Analysis of Weighted Directed Graphs

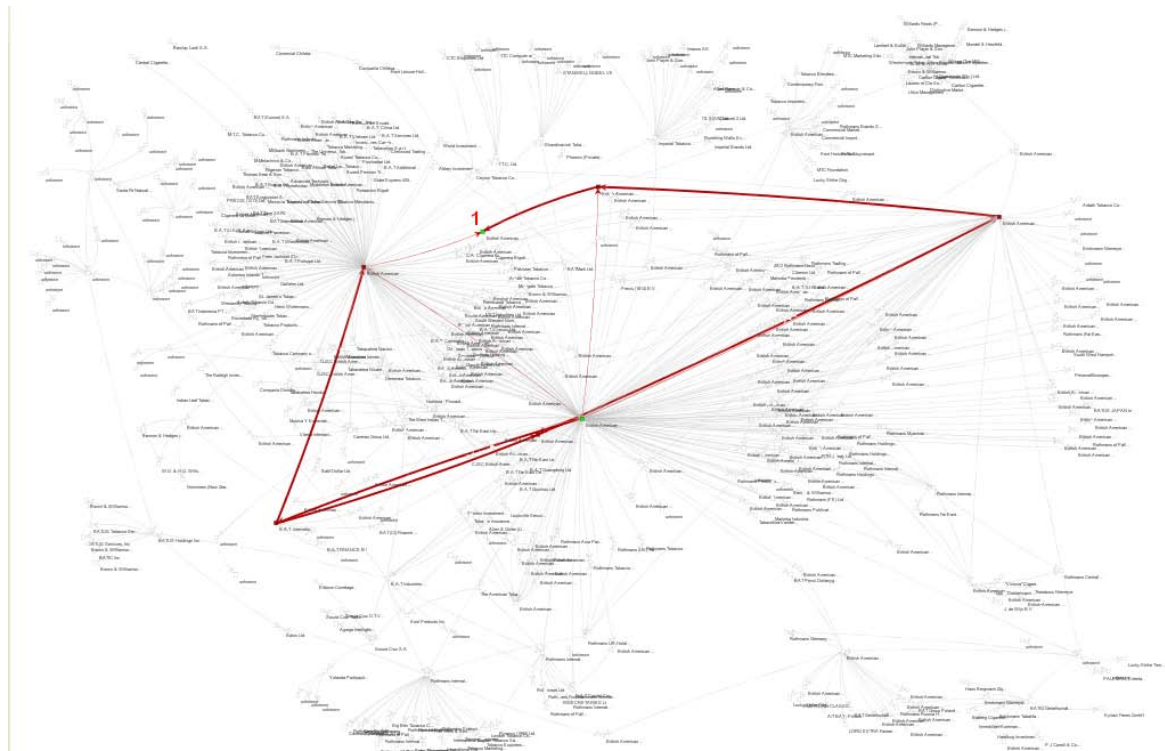


Figure 3.64.: Control structure of British American Tobacco (Nigeria) Ltd. (1).

The complex holding relationships of the British American Tobacco group are clarified when looking for *extended double-cross motifs* (i.e., substructures including several companies holding shares in the identical set of other companies), which turn out to be pervasive in the ownership structure. This complexity reflects the ample geographical diversification the group. One of the many holders in the double cross motifs is the root node (British American Tobacco plc.) showing its dominant position in the structure, although directly holding only a minimal participation (see Figure 3.65).

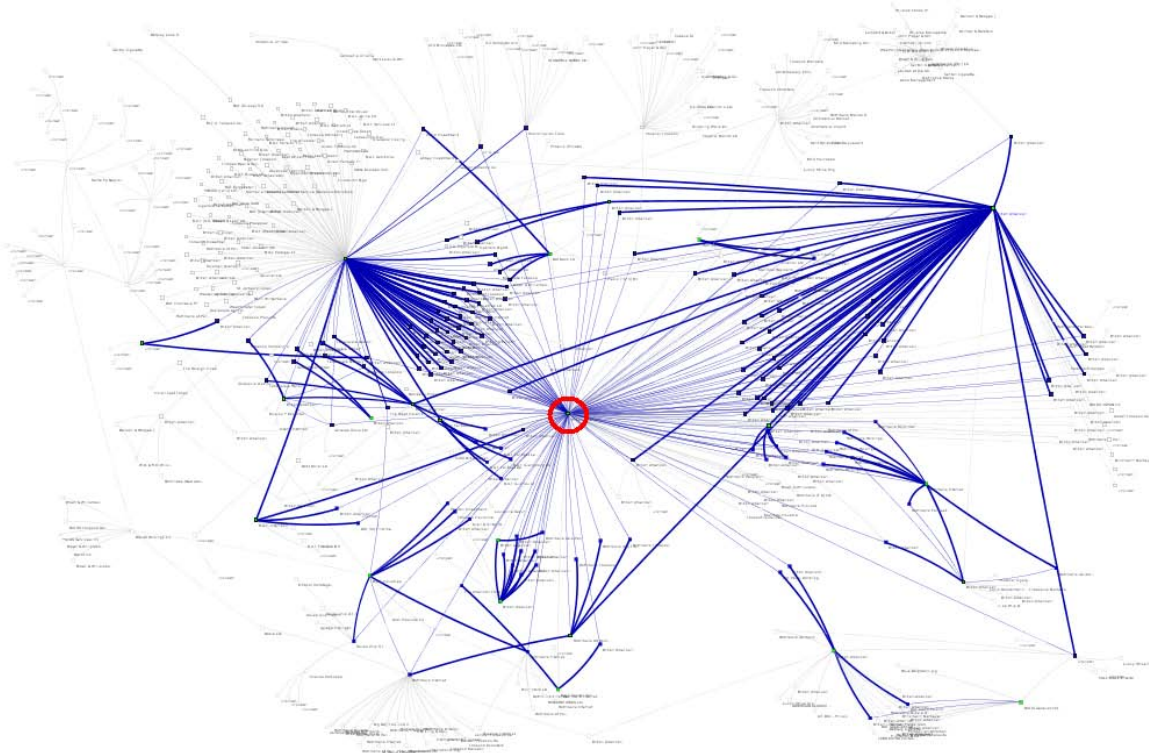


Figure 3.65.: Extended double cross motifs in the structure of British American Tobacco plc. showing the important position of the company BAT plc. in the company structure with connections reaching in all the corporate parts. The company is emphasized with a red circle.

A number of *outgoing star* motifs can be seen as being “on the periphery”: of the group. Figure 3.66 shows that in many cases intermediate sub-holding companies are involved. This is the case for the German operations of British American Tobacco (Hamburg International) GmbH, which is the sub-holding company for some of the operations in Eastern Europe – the hub in the red box – and the operations in Serbia, Poland, Slovakia, Romania, etc. are the spokes. However, an interesting aspect of the ownership structure is that the three main companies (the root node and the two controlling companies) also hold direct participations in many “spoke” companies at the end of the chain, which at closer inspection are identified as individual country operations, often in mature markets such as Switzerland, France, Italy.

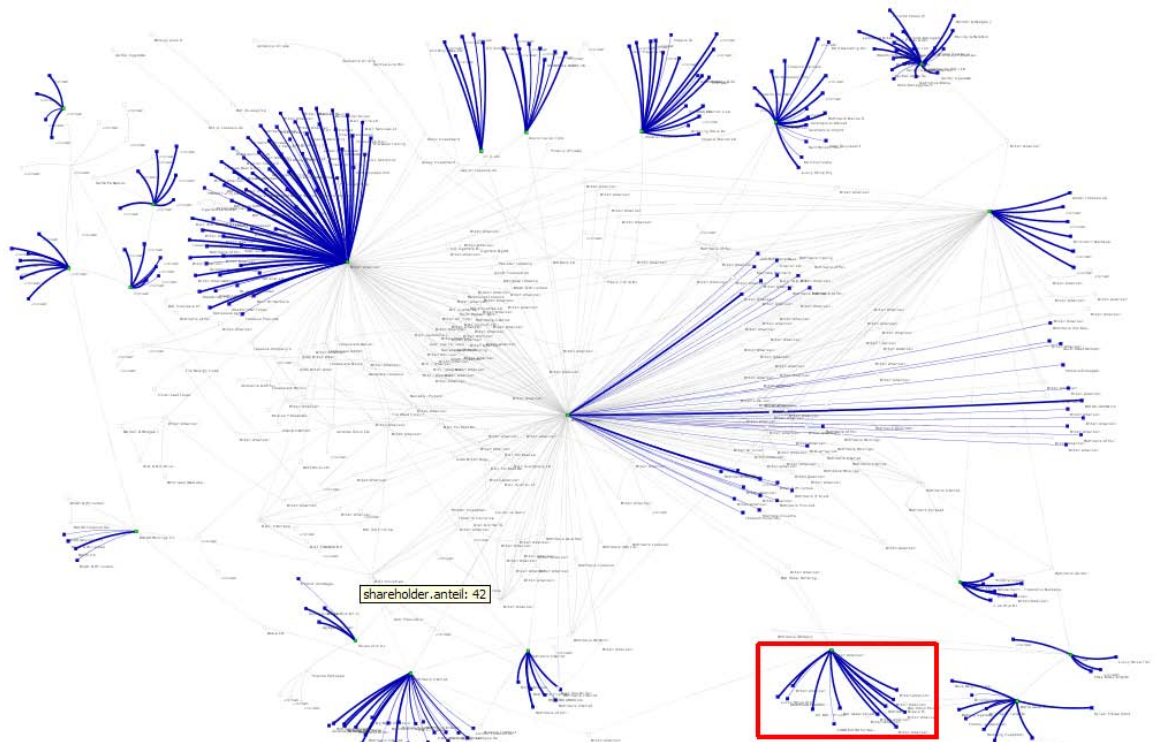


Figure 3.66.: The out-star motifs showing companies at the end of the shareholding chain. Highlighted in red box is are the subsidiaries of British American Tobacco (Hamburg International) GmbH – holdings in Eastern Europe.

Visual “What-if-analysis” of Shareholding Structures shows what impact a change in the network has on the local structures. For example, a financial analyst might be interested to investigate what happens if a company defaults (i.e., is closed) or opens (i.e., is created) or what happens if company A buys or sells shares in a company B. One option is to analyze impact on direct and indirect shares as supported in the exploratory analysis (see above) or to analyze impact on local relationships.

For the BAT corporation, we simulate the closure of the company Rothmans International Holdings II B.V. (RIH), which is a holding company for many BAT operations in the UK. BAT bought Rothmans in 1999. The Rothmans part of the group and the connection between Rothmans Int. Holdings and BAT can be seen in the Figure 3.67.

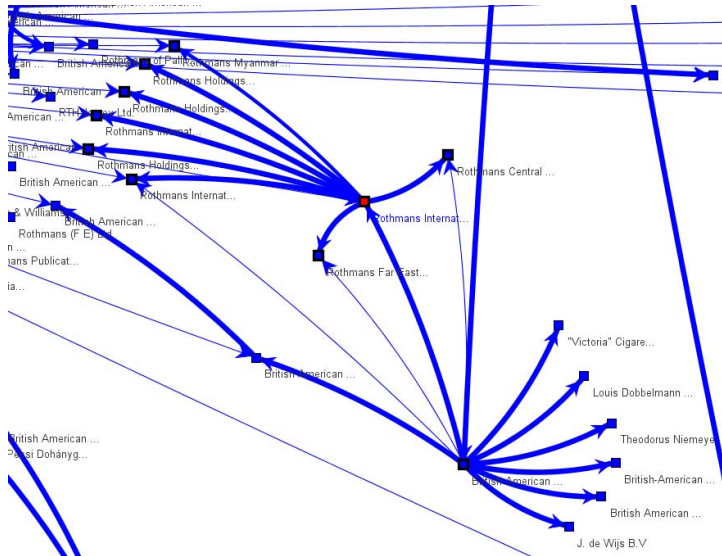
In case of closure of RIH, two alternative scenarios can be investigated. Either the mother holding company (British-American Tobacco Netherlands B.V.) decides to buy shares in some of the RIH’s companies or not. Both cases are shown in the Figure 3.68. Interestingly, owing to the special shareholding structure, in both cases the companies held by RIH stay in the group as all of them are held either by the ultimate shareholder BAT directly or by the BAT Netherlands. Nevertheless, the holding relationships of these companies change (see Figure 3.69 as an illustration of shareholding relationship with Rothmans Central and Eastern Europe before and after the change).

The motif analysis shows that the company Rothmans International Holding was the center of an outgoing star (having many subsidiaries) and at the same time, as a subsidiary of BAT plc. and BAT Netherlands, part of an outgoing motifs centered around BAT plc. and BAT Netherlands (see Figure 3.70 left). After the redistribution of the participations, the out stars centered on BAT plc. and BAT Netherlands change their shape – now including the subsidiaries of RIH (see Figure 3.70 right). In contrast, the feed-forward motifs which RIH was part of are not maintained in the new structure (see Figure 3.71). This shows the significant role of Rothmans Int. Holding for the structure in this part of the BAT group.

Aggregation of Shareholder Structures using Motifs can be used in order to simplify the network and reveal relationships at a higher level of abstraction. In our example, in a first step, all “leaf out stars”, i.e., such outgoing stars with leaf nodes, are aggregated. It should be kept in mind that the nodes at the center of an out-star are usually holdings controlling many companies. 49 leaf-out stars at the end of the shareholding chain (see Figure 3.72) can be discerned and are affected by the aggregation. This procedure does not affect relationships forming the other (non-star shaped) motifs, but the nodes within the motifs change: The nodes in the center of the star become aggregated nodes (the blue circular nodes). Overall, the aggregation reduces the graph order from 532 to 233 nodes. This strong reduction renders particularly visible the out-star shape of the entire corporate structure.

The visualization of feed-forward motifs (see Figure 3.73 left) reveals that around half of the companies (nodes) at the top of the feed-forward substructures are also at the center of stars (the green colored nodes). The caro motif (another type of structure supporting higher control in a company via third companies) also shows that top companies (apart from BAT Industries plc. highlighted in red circle) in such substructures are often also centers of caro motifs (see Figure 3.73 right). This confirms the previous analysis findings that there are several holding companies who have a controlling position in the network as “satellites” of the ultimate shareholder BAT plc..

After the first aggregation round, the graph still seems very complex and difficult to interpret. In a further aggregation round all feed-forward motifs which allow for higher control in the network are now condensed. Interestingly, the feed forward motifs were concentrated around the ultimate shareholder (BAT plc.). After this second aggregation (see Figure 3.74) the network has become simpler (the number of nodes decreased further to 62 nodes from previously 233). Although the complexity of the graph has strongly been reduced, it is still



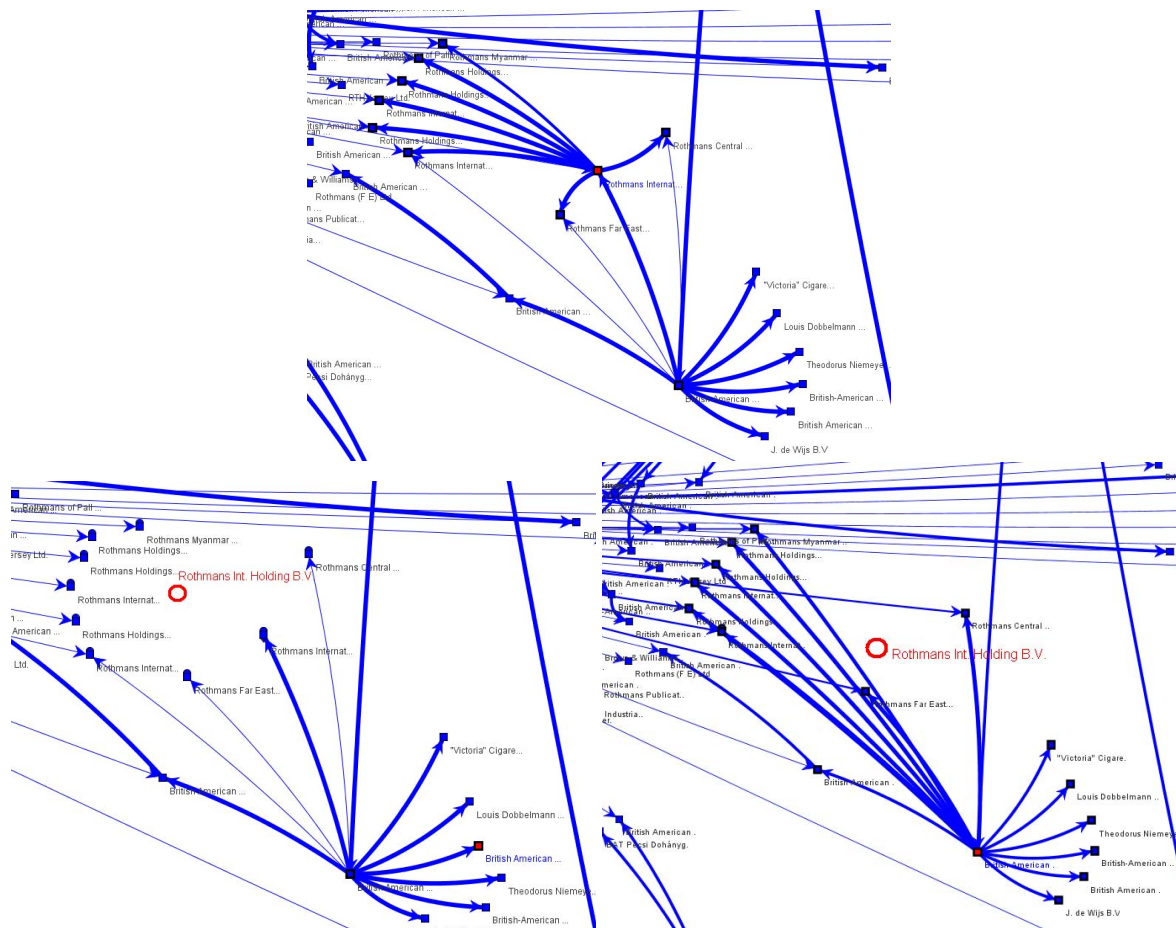


Figure 3.68.: The two cases of changes of shareholding relationships after this simulated closure of the company Rothmans International Holdings (RIH). Top: The shareholding structure showing the situation before closure where the company RIH holds directly shares in 8 companies. Bottom left: The shareholding structure showing the situation after closure when no redistribution of shares takes place. The companies formerly co-held by RIH are now held only by the co-shareholders whereby they stay connected in the network. Bottom right: The shareholding structure showing the situation after closure in case that the mother company (BAT Netherlands) buys shares in subsidiaries of Rothmans Int. Holdings N.V..

3. Visual Analysis of Weighted Directed Graphs

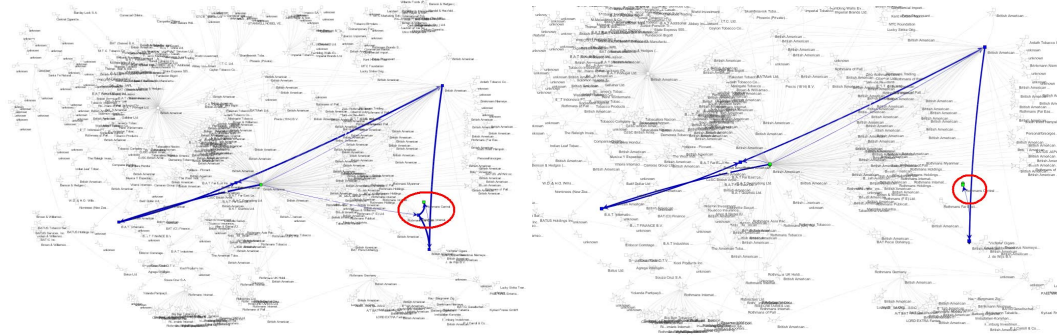


Figure 3.69.: The relationship between Rothmans Central and Eastern Europe and BAT plc.. Left: The shareholding situation before the simulated closure of Rothmans International Holding B.V.. Right: The shareholding structure after the simulated closure.

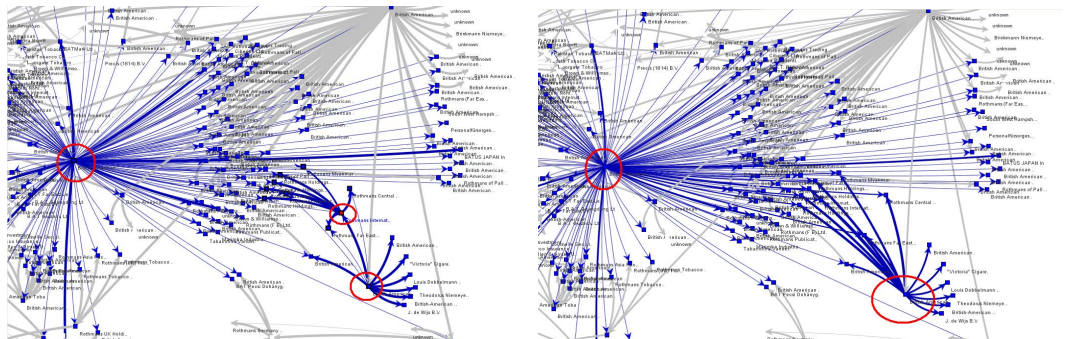


Figure 3.70.: The impact of the simulated closure of Rothmans International Holding on local substructures – outgoing star motifs. Left: The outgoing star motifs before the closure. Right: The changed outgoing motifs after the closure showing that one motif disappeared and new edges are included in the motifs.

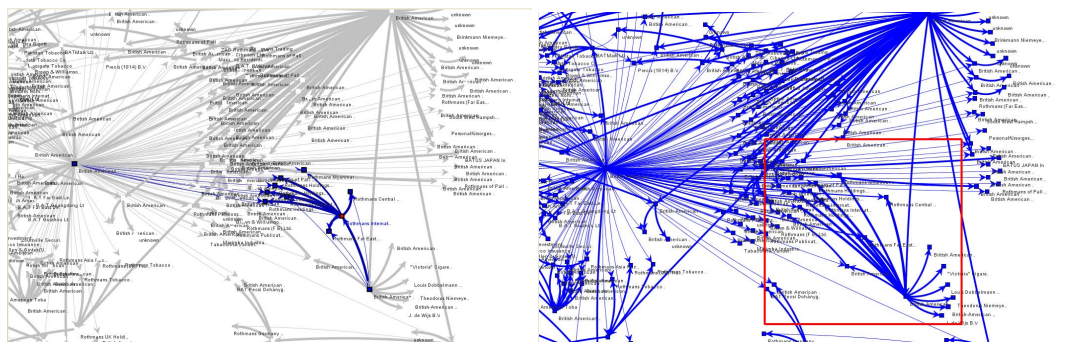


Figure 3.71.: The impact of the simulated closure of Rothmans International Holding on local substructures – feed-forward motif. Left: The feed forward motifs before the company closure. Right: Feed-forward motifs after the closure with implications throughout the network.

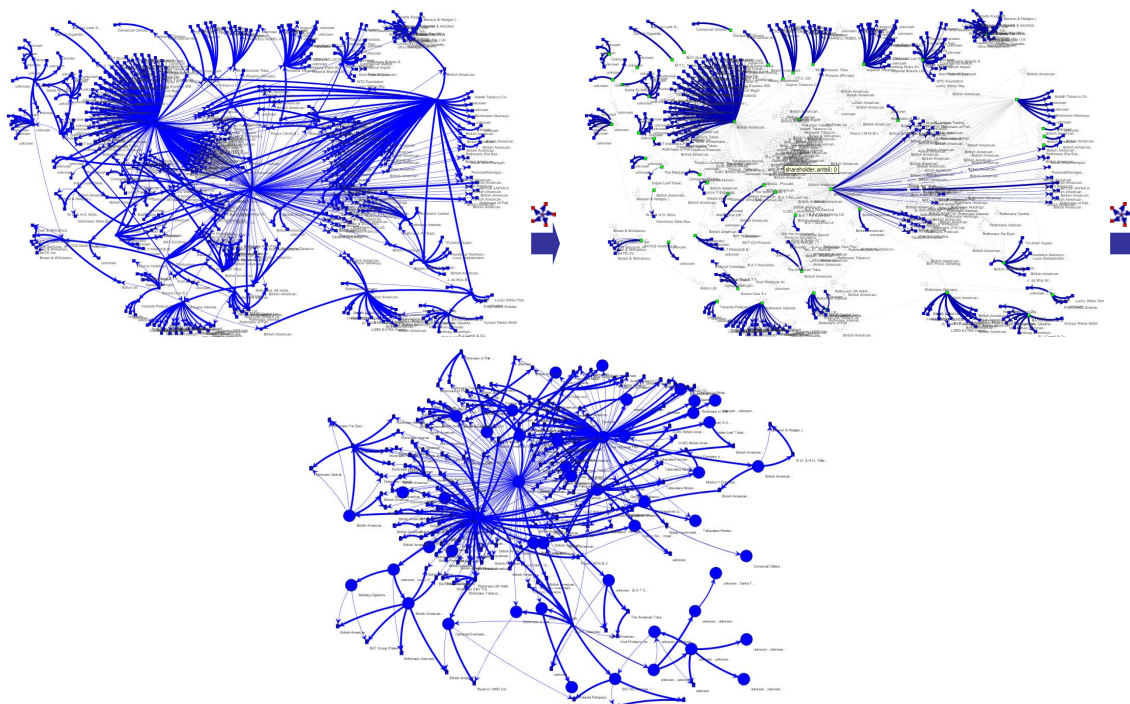


Figure 3.72.: Example of motif-based aggregation of the BAT corporation. Top left: Original network. Top right: Leaf outgoing star motifs to be aggregated are highlighted. Bottom: Resulting aggregated graph with blue circular nodes indicating the aggregated nodes. Each aggregated node includes one outgoing star motif.

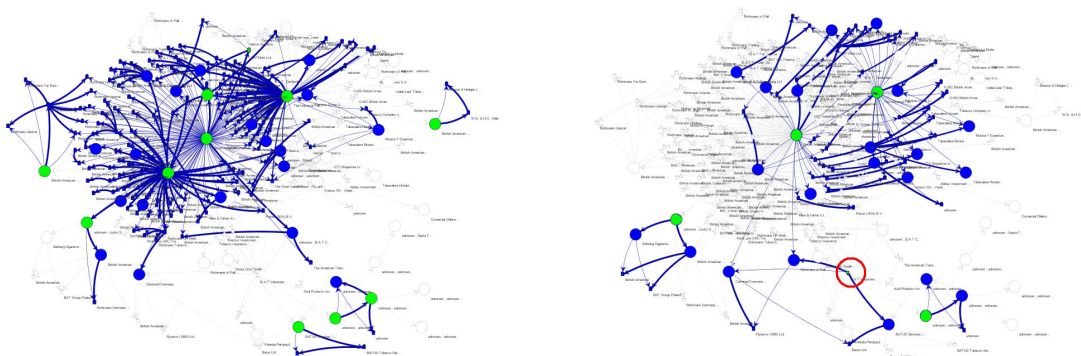


Figure 3.73.: The motifs found in the aggregated network. The motifs which include also the aggregated nodes (circular nodes) show the structural relationships between motifs in the original network. Left: Feed forward motifs. Right: caro motifs found. The green nodes highlighted the top companies in the motifs. In both cases, many aggregated nodes form the top parts of the motifs. An exception is company BAT Industries highlighted in red circle.

able to represent the main structure of the corporation. The main ultimate shareholder in the middle is at the top of a feedback motifs to which further star shaped companies are connected (see Figure 3.74 top left). The strong interconnectedness in the network is also seen in the newly created feedback and reciprocity motifs (see Figure 3.74 bottom). Further inspection of the aggregated network shows that the company BAT Investments plc. is member of all motifs which may be an interesting fact for the analysts initializing further deeper analysis of the workings of this company and its central position within the group as the financial hub.

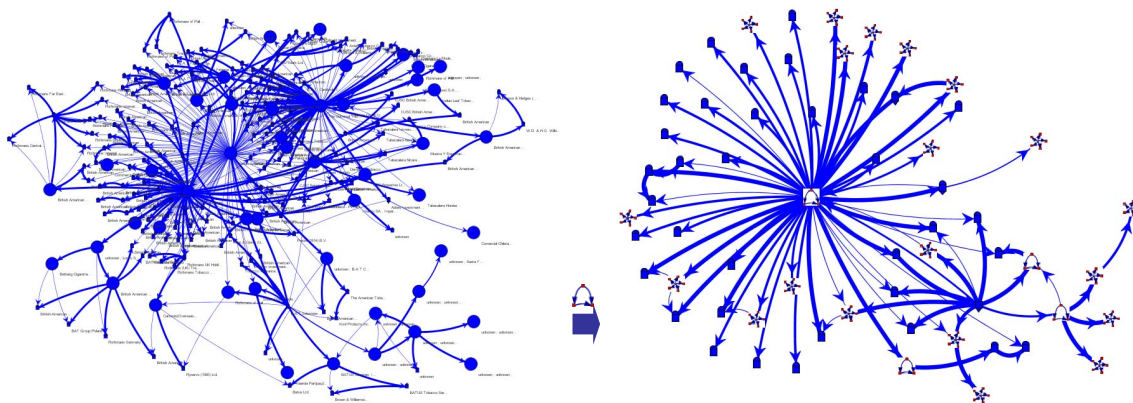


Figure 3.74.: Hierarchic/multiple motif-based graph aggregation on the example of the BAT network. Left: Aggregated graph before the second aggregation showing circular aggregated nodes. Right: The simplified graph on a higher abstraction level resulting from second aggregation using feed-forward motif. The icons represent the aggregated nodes and display the latest type of aggregation used for creating the node.

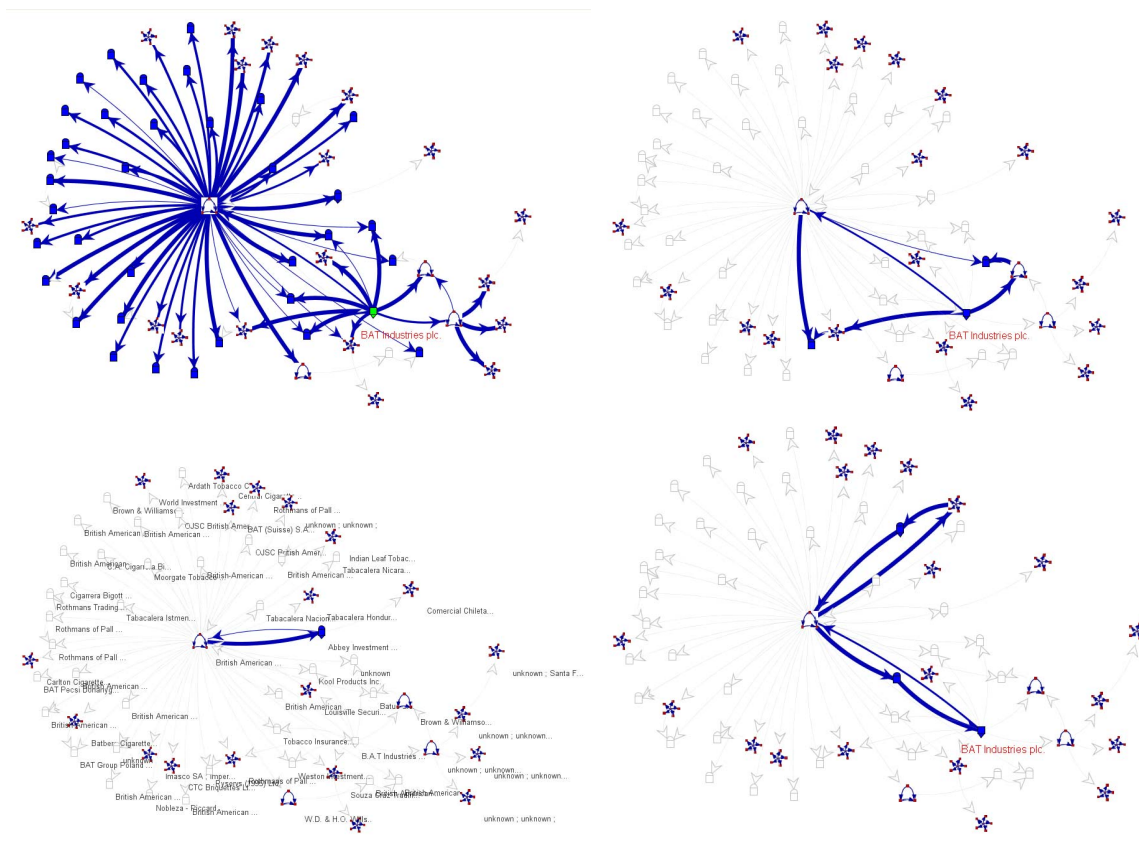


Figure 3.75.: Motifs in the twice aggregated BAT network (see Figure 3.74). Top left: Outgoing stars. Top right: caro. Bottom left: reciprocity. Bottom right: feed-back. The motif analysis shows the importance of the central node being involved in all motifs. This central node results from aggregation in both steps.

3.7.5. Visual Analysis of Shareholder Networks using SOM Clustering

In this section, we show how SOM-based graph clustering can be applied to the *analysis of types of company structures formed in an economic system*. Each shareholding structure in the economy forms a graph (weakly connected component in the whole shareholding network). In the whole economy a large number of such networks exist, which are difficult to compare and thus complicate the analysis of the structural composition of the whole economy. Therefore, we use clustering in order to analyze the individual company networks, and thereby help to answer analytical questions such as: *Are there corporate shareholding structures typical to the studied economy? Which types of structures are exceptional? How frequent are particular structures?*

In our approach, we employ feature based graph similarity for clustering. Therefore we first explain how features introduced in Subsection 3.6.2 apply to shareholding networks. Clustering results using various feature combinations are described in the following subsection.

3.7.5.1. Features for Comparison of Shareholder Networks

We shortly explain selected features interesting for clustering analysis of shareholding networks.

1. **General features** The size of network shows whether the corporate holding is large or small, the degree of completeness shows whether there are complex relations in the network and average degree is a measure of strength of controlling rights in the network.
2. **Reciprocity features** the degree of cross-holdings and “close relationships” between companies.
3. **Distance features** measure the “size of the pyramid structure” – the size of the chain of shareholdings in the network.
4. **Clustering features** measure the closeness of relationships between companies.
5. **Degree distribution features** show the distribution of the number of companies held by a company, or number of companies holding shares in one company, whether there are companies holding shares in themselves etc.
6. **Motif-based features** are explained in the Subsection 3.7.4.1 on page 109.

3.7.5.2. Results

For the analysis of the German shareholding data set taken from the Amadeus database, we rely on interactive feature selection while using a constant SOM grid size of 9×12 and the same clustering parameters. Changing of feature sets in combination with SOM clustering leads to various views on the data set. These views show the distribution of types of components. It shows which components are frequent in the data set and which are exceptional under the given feature set. In the following paragraph, our findings from the shareholder data set are presented.

The SOM grid (see Figure 3.76) shows that the shareholding components in Germany vary from simple 2 node graphs (bottom left corner) to more complex larger graphs (top right corner). The star-shaped graphs are the most important corporate structures, having the highest frequency and occupying multiple cluster centers (with varying graph sizes). The member views allow detailed analysis of individual companies with similar structures.

Figure 3.77 shows a SOM produced using only the *number of nodes* as a graph descriptor as a first naive approach. From top-left to bottom-right, graphs of increasing size are arranged on the map. It already reveals what sizes of shareholder structures there are in the dataset. Subgraphs with up to 6 nodes are very frequent. Then, with a gap, larger graphs occur.

Figure 3.78 shows a map obtained by extending this feature by the *graph completeness*. In effect, the initial coarse SOM layout is refined by accommodating more differentiation regarding a notion of the graphs' complexity. The result shows that the larger the company structures are the more complex is the relationship within them. Small holdings consist mainly of many shareholders of one company and larger include more interwoven cross-holdings.

The inclusion of the distance feature (see Figure 3.79) reveals an interesting pyramid-like structure with 5 levels (shareholder "chain").

Finally, Figure 3.80 shows the usefulness of the feature controlling for *number of loops* in the graph for extracting extraordinary examples of companies directly holding shares in themselves. This phenomenon is unexpected. It can be either an outlier, a data quality problem or an interesting company structure which should be reflected in the subsequent detailed analysis.

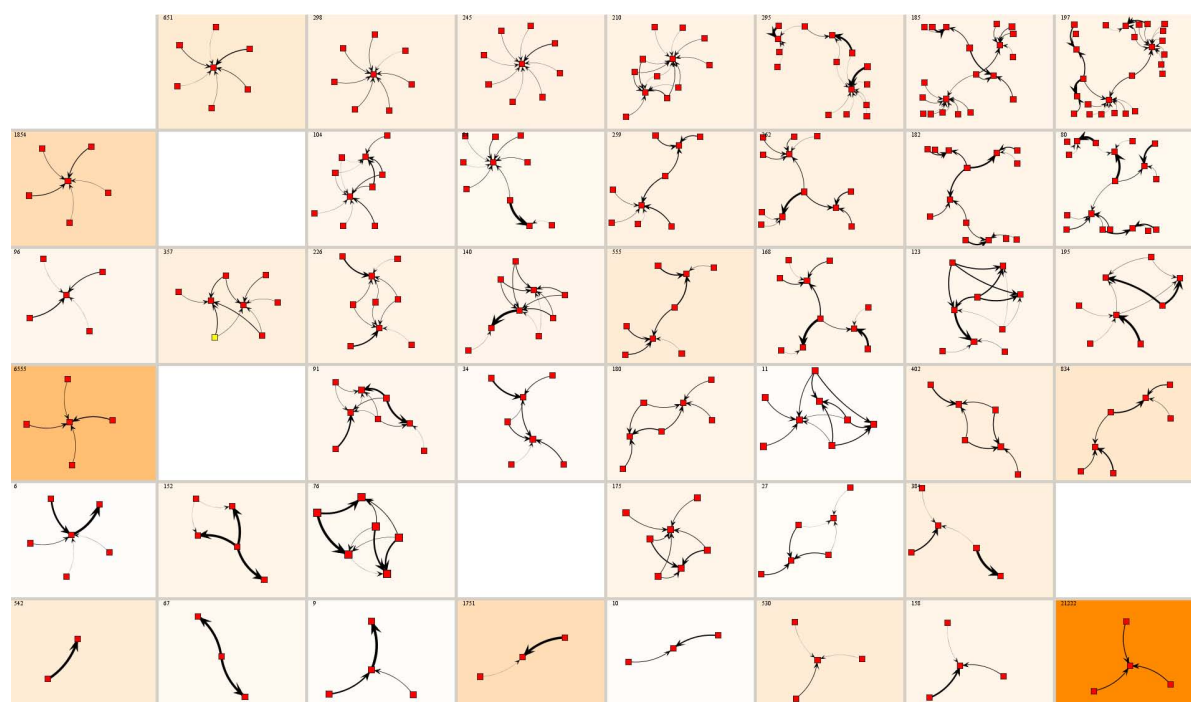


Figure 3.76.: Graph clusterings obtained for multiple features with a larger weight of the number of vertices feature. Each SOM cell contains the nearest neighbor graph, while the background color indicates the frequency of the cell elements in each cell. It shows the general distribution of German company structures with predominant star-shaped structure and high frequency of small networks up to 5 nodes.

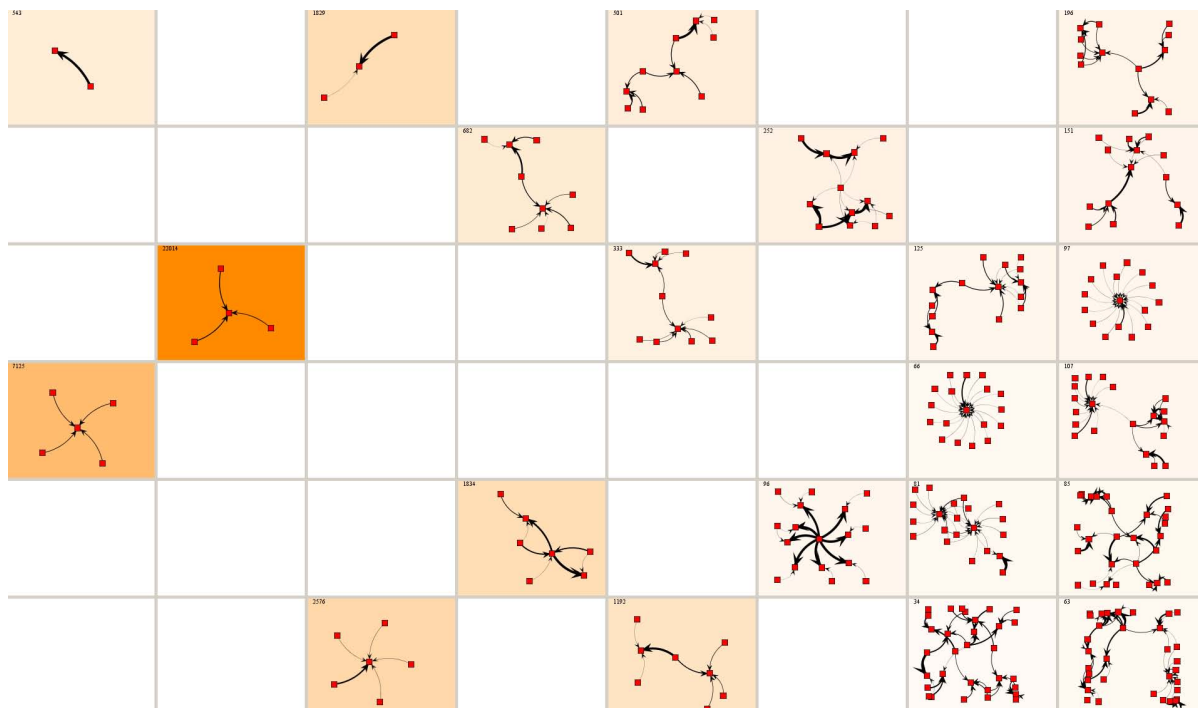


Figure 3.77.: Graph clustering obtained for the number of vertices graph feature selection. Each SOM cell contains the nearest neighbor graph, while the background color indicates the frequency of the cell elements in each cell. It shows the distribution of German shareholding networks according to number of companies from smallest 2 company networks in upper left corner to larger networks on the right. The background color shows the large number of smaller companies.

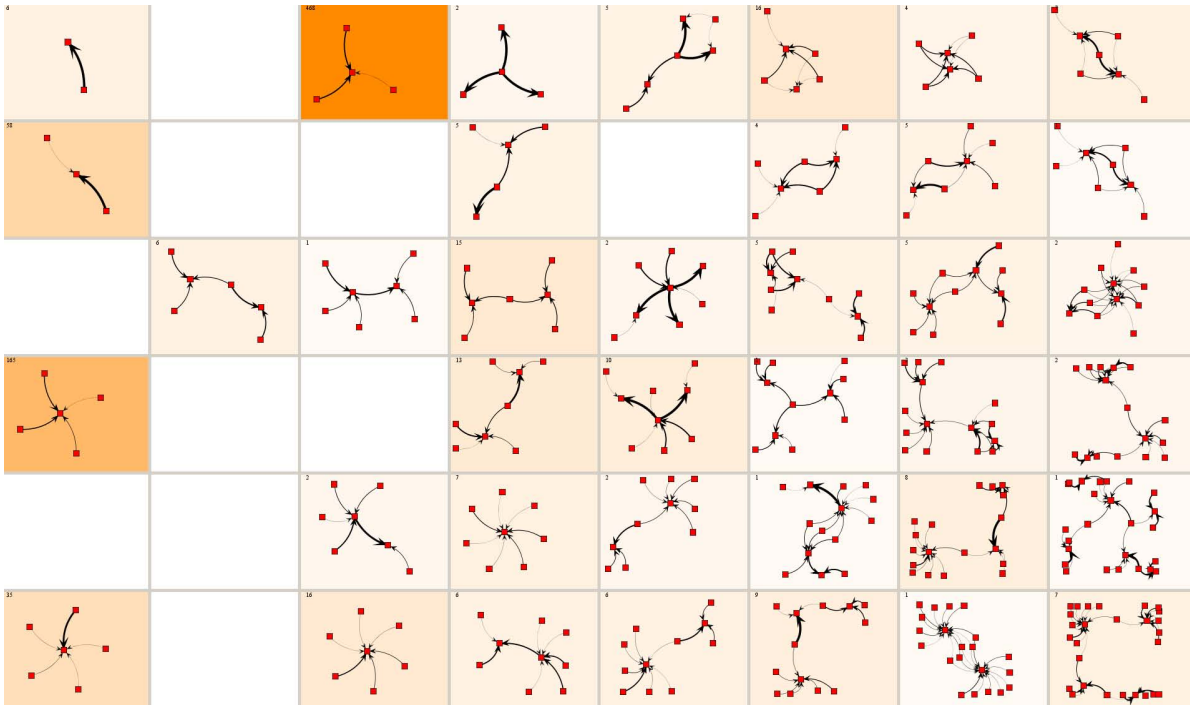


Figure 3.78.: Graph clusterings obtained for the number of vertices and completeness graph feature selection. Each SOM cell contains the nearest neighbor graph, while the background color indicates the frequency of the cell elements in each cell. It shows the graph structure distribution with regard to both number of companies and complexity of their relationships. This distribution is a refinement of the previous SOM clustering using only number of vertices (see Figure 3.77).

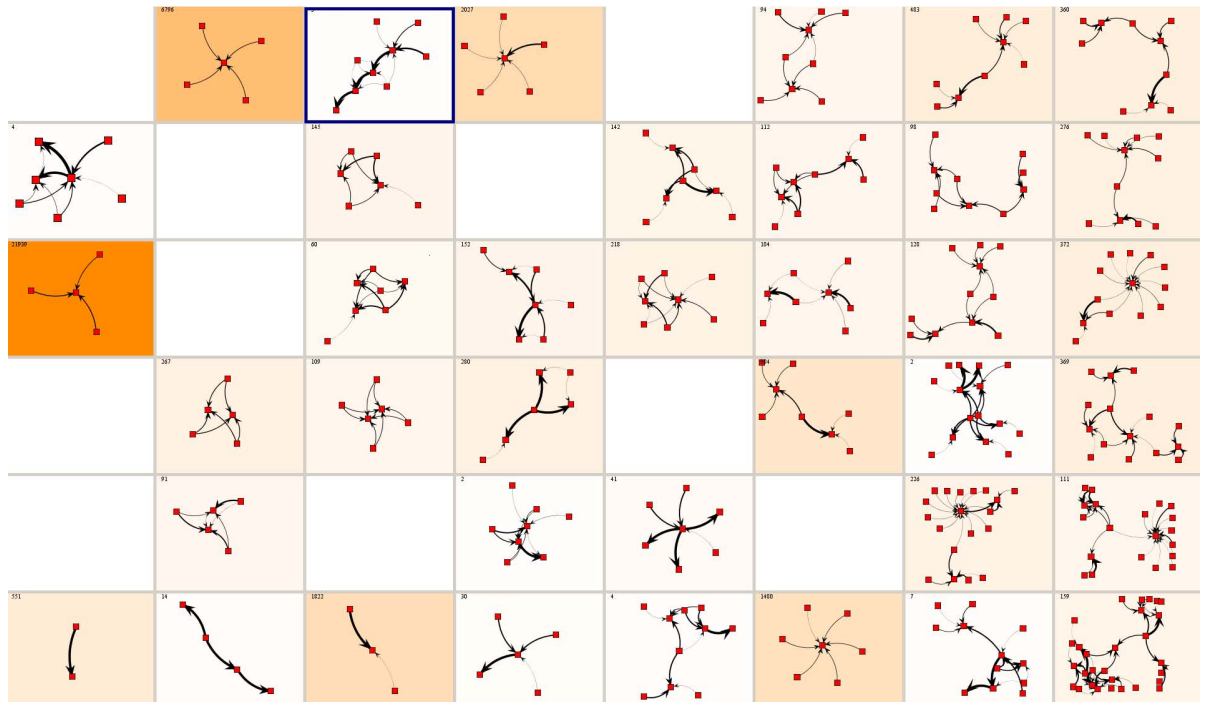


Figure 3.79.: Graph clusterings obtained for features including the number of vertices, completeness and distance features. Each SOM cell contains the nearest neighbor graph, while the background color indicates the frequency of the cell elements in each cell. It reveals structures with “longer holding chains” (the SOM cell is highlighted in blue).

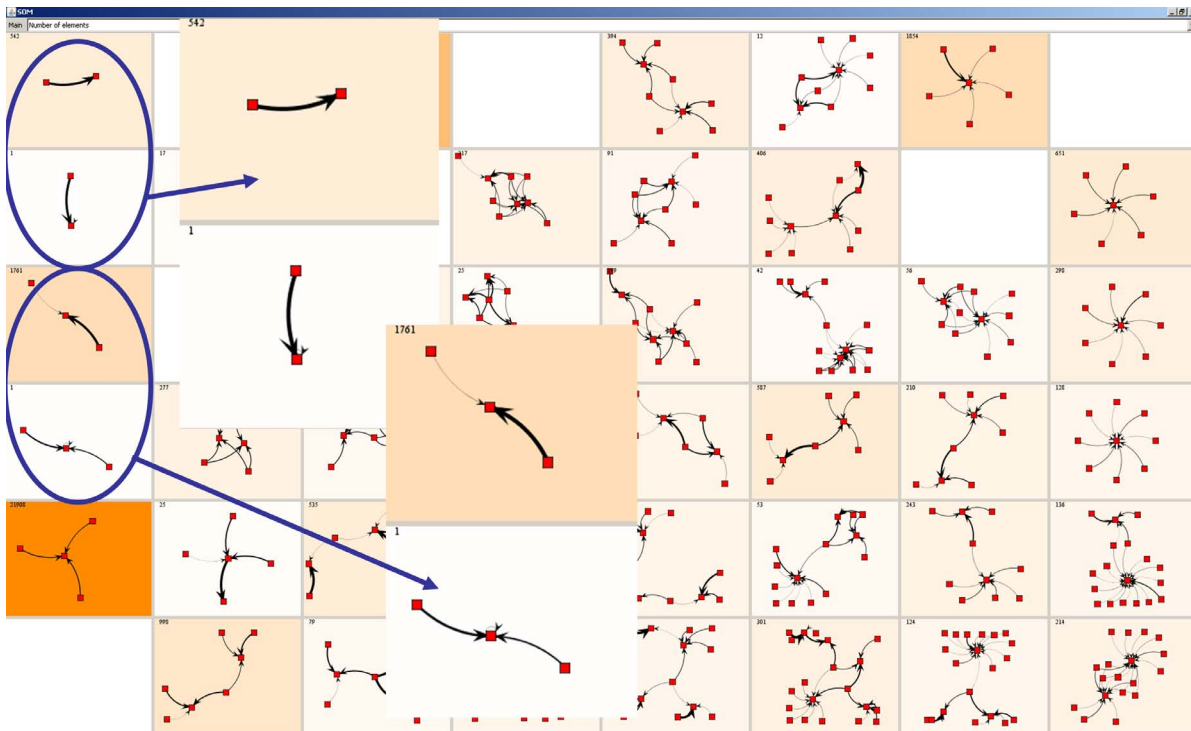


Figure 3.80.: Graph clusterings obtained for various selected features including the number of loops. Each SOM cell contains the nearest neighbor graph, while the background color indicates the frequency of the cell elements in each cell. The inclusion of the feature controlling for loops reveals two types of extraordinary company structures (see zoom ins) where one company holds shares directly in itself. This can be an interesting fact or a problem in the original data collection.

4. Visual Analysis of Two-Dimensional Time-Dependent Data

4.1. Introduction

The analysis of two-dimensional time-dependent data¹ plays a prominent role in many applications such as transportation, meteorology, finance or biology. For example in the financial domain, the analysis of data on risk and return dynamics is essential for financial investment decisions.

Two-dimensional data stem either from direct measurement or from simplified projection of multi-dimensional data into two dimensions (e.g., by PCA, MDS or Sammon's mapping algorithms).² These data sets are often time-dependent (e.g., registered repeatedly in uniform time steps over a period of time). Moreover, the entities for which data is analyzed can often be grouped into several classes (e.g., assets by country).

The analysis of two-dimensional time-dependent data can be regarded as an instance of the *trajectory analysis* problem treated in spatio-temporal research. Respective work in that area deals with movement data observed in real-world coordinates [AAW07, CZQ*08]. There are two main *differences between analysis of geo-based trajectory data and general abstract data* (e.g., financial). Firstly, geography-based data often need to be analyzed using fixed coordinates. Abstract data, by contrast, do not have an inherent position and can therefore be more flexibly normalized, translated or rescaled depending on the analytical problem to be solved. Secondly, movement data are often measured in irregular and/or unequal number of time steps (e.g., when car passes a sensor), while the abstract data that is going to be analyzed in this chapter, is measured at regular intervals (e.g., each working day in a week)³.

There are *two major approaches* to the analysis of two-dimensional time-dependent data. The first approach uses appropriate visual techniques for the visual exploration of the data. The visualization techniques are usually based on scatterplots extended with animation or trajectory visualization in order to capture data dynamics. The second approach applies data mining techniques for finding interesting patterns in the data. It considers the analysis and description of important properties in the data. Of primary concern are methods to define appropriate similarity functions to query, compare, and cluster trajectories [AA07, PKM*07], and to support the detection of interesting patterns [PBKA08a]. “However, spatio-temporal data mining is still in its infancy, and even the most basic questions in this field are still largely unanswered: what kinds of patterns can be extracted from trajectories? Which methods and algorithms should be applied to extract them? One basic data mining method that could be applied to trajectories is clustering, i.e., the discovery of groups of similar trajectories.” [NP06]. Recently, unifying both approaches, Visual Analytics research (e.g., [IWSK07, AAR*09]) has focused on the

¹The terms two-dimensional time-dependent data, two-dimensional time series, two-dimensional dynamic data or two dimensional dynamic points are used interchangeably in the following.

²The projection often leads to loss of information and thereby lower accuracy for the interpretability of the data. As we assume the two-dimensional data as given this aspect is disregarded in the following.

³The data measured each working day may also have individual missing data owing to holidays leading to irregularities in time steps. We assume the data to be cleaned for such problems.

combination of both interactive visualization and data mining to create visual analysis systems dealing with two-dimensional time-dependent data.

4.1.1. Tasks

The analysis of two-dimensional time dependent data concentrates either on the analysis of dynamics of individual data items or groups of data elements (grouped, for example, by their class information). The investigation often aims at gaining insights into the general dynamics of the data and strives to discover interesting patterns in the data. Against this background, this chapter deals with the following three main analytical tasks (see also Figure 4.1):

1. exploration of *individual entities*' dynamics on the basis of two-dimensional time dependent data,
2. analysis of *grouped entities*' dynamics on the basis of two-dimensional time dependent data,⁴
3. analysis of patterns in *individual entities*' dynamics on the basis of two-dimensional time dependent data.

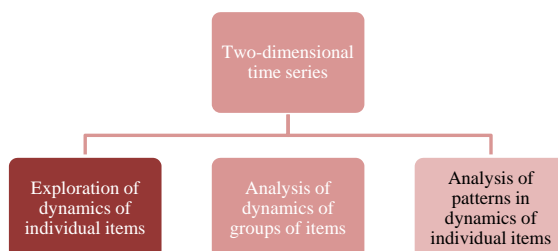


Figure 4.1.: Tasks in the analysis of two-dimensional time dependent data.

These three tasks are non exclusive and are often combined in the real-world analytical process. For example, the analyst may first like to gain an overview of the data by exploring the general dynamics of the individual items and then like to extract important data patterns in more detail. Afterwards she may wish to explore only the part of the dataset which includes such important patterns (see Figure 4.2a). Alternatively, the analyst may first analyze the movements of groups of data items, and then concentrate on the dynamics of individual entities within the groups and finally compare the patterns found in each group (see Figure 4.2b). What analytical process is pursued depends mainly on the analytical goal, user's experience, found insights during the analysis and other factors.

4.1.2. Contribution

For each of the three above-mentioned analytical areas, we propose a suitable approach. These approaches can be combined flexibly during the analysis process.

Firstly, for visual exploration of two-dimensional time dependent data, two interactive visualization methods based on scatterplots: animation and trajectories are discussed. In connection with animation, it should be recalled that its effectivity is highly dependent on the human perception capabilities. Against this background, we have conducted a perception study and present its results. These results give guidance for application of animation in the visualization of two-dimensional time dependent data.

⁴We assume that the grouping stays constant over the analyzed time period.

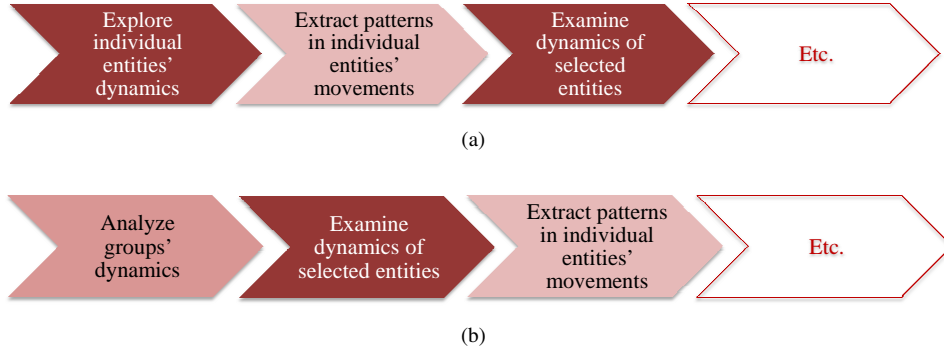


Figure 4.2.: Two examples of possible process in the analysis of two-dimensional time dependent data.

Secondly, turning to the analysis of groups' dynamics for this type of data, we extend the interactive visualization with a feature extraction procedure. We thereby create a visual analytic tool that allows to spot interesting developments in the data. We extend the set of features for analysis of static data distributions, with new features suitable for analysis of time-dependent data. We consider thereby three analytical cases in the assessment of data dynamics: a) entities within groups, b) individual groups, and c) relationships between groups.

Finally, identification of patterns in the two-dimensional dynamic data is enhanced by an interactive visual analysis tool based on SOM clustering. It provides visual output for exploration of the results and allows to control the clustering process.

4.1.3. Chapter Overview

Section 4.2 discusses related work in the area of visualization of (dynamic) two-dimensional points and their algorithmic and visual analysis. Afterwards, in Section 4.3, we present our three approaches for the visual exploration and analysis of two-dimensional dynamic data. The detailed description of these techniques is provided in Sections 4.4, 4.5 and 4.6. The application of our methods focuses on the analysis of dynamic risk-return data from the financial investment domain (see Section 4.7).

The work presented in this chapter is partially based on the following publications [TK07], [TSPK08], [TS08], [vLBRS09], [KTSZ08], [STFK07], [SBTK08], [TSB*08] and [SBvLK09].

4.2. Background

This section presents the main techniques for visual analysis of two-dimensional time series relevant to the tasks presented in the introduction (see Subsection 4.1.1). We firstly define two-dimensional time series (with regard to both individual and group dynamics). In the following subsection, we overview algorithmic techniques related to analysis of two-dimensional dynamic data. We then proceed with visualization techniques for this data type. In turn, visual analysis techniques combining interactive visualization and algorithmic analysis for analysis of this data are presented.

4.2.1. Definitions

An *individual two-dimensional time series* (i.e., 2D point time series, a dynamic 2D point) can be regarded as an instance of *trajectory* – the term used mainly in geographic area. In this respect, “A trajectory is the path made by the moving entity throughout the space where it moves. The path is never made instantly but requires a certain amount of time.” I.e., “Trajectory = space-time path” connecting individual measures time-space points. [AAPS08], where linear movement of the points with constant speed between time points is assumed. Along these lines, we define *trajectory of an entity k* : T^k as:

$T^k = \{t_0^k, t_1^k, \dots, t_n^k\}$, an ordered set of points, where

t_i^k is the 2D position $t_i^k = [x_i^k, y_i^k]$ of the entity k at time point i , where

$i \in I, I = \{0, \dots, n\}$.

In this example, t_0^k is the starting point and t_n^k is the end point of the trajectory T^k .

(see also Figure 4.3 for an illustration).

Time points $i \in I$ can be equidistant (e.g., once an hour or once a day) or irregular (e.g., each time an object passes a sensor). In our work, we concentrate on trajectories measured on uniform time intervals.

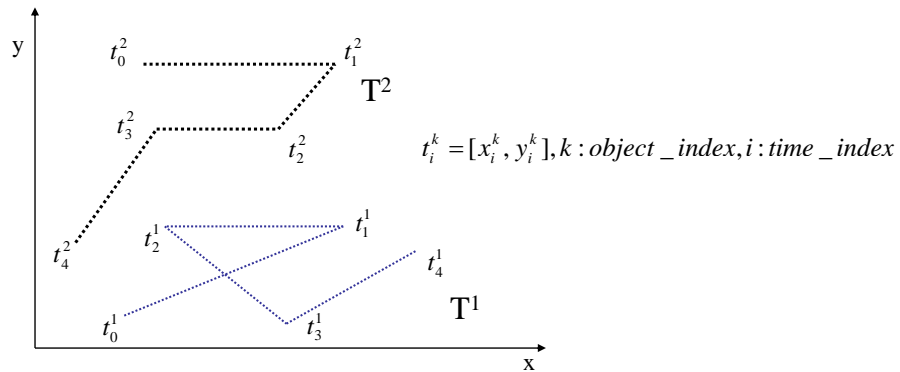


Figure 4.3.: Example of two-dimensional time series (trajectories). The trajectory points t_i^k are connected by a line forming a trajectory path.

In addition, we can define a *trajectory fragment* [AAPS08] as a part of the whole trajectory as

$$T_F^k = t_{i_0}^k, \dots, t_{i_n}^k; T_F^k \subset T^k, \text{ where} \\ I_F = \{i_0, \dots, i_n\}, I_F \neq \emptyset \text{ is a nonempty consecutive part of original time space } I \\ I_F \subset I.$$

A *fragment of a trajectory* is usually created by selecting a specific time subset from the whole time period. In analogy to the fragmentation (i.e., partitioning) of a trajectory into individual fragments is done by dividing the whole time set I into disjunct time subsets

$$I_{F_0}, \dots, I_{F_l}, \text{ where} \\ \bigcup_{j=0}^l I_{F_j} = I \text{ and} \\ I_{F_j} \cap I_{F_h} = \emptyset, \forall j, h \in 0, \dots, l; j \neq h.$$

In our work, we concentrate on fragmentation (partitioning) of a trajectory into a set of disjunct trajectory fragments (subtrajectories) with equal number of time points in each fragment.

If entities (also referred to as objects or items), for which we define the trajectories, are *grouped* (e.g., by a specific attribute such as country of origin) (see Figure 4.4), then the time development of each *group of entities* T^G creates a complex composition of movements of individual group members $g \in G$ (see Figure 4.5 for an illustration). This is stated in [AAPS08] as: “*The collective movement behavior of a population of entities over a time period is a complex configuration built from movement characteristics of all entities at all time moments, which has no arrangement with respect to the population of entities and has a continuous linear arrangement with respect to the time.*” (see also Figure 4.5).

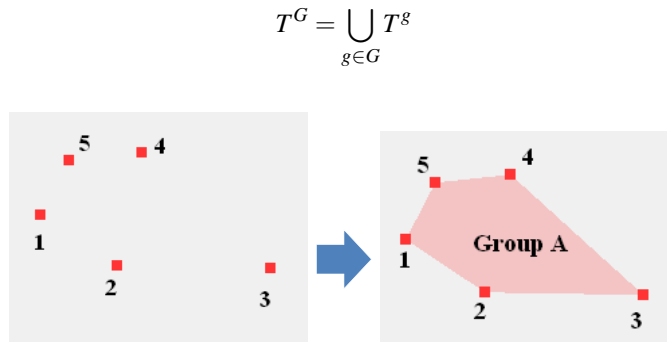


Figure 4.4.: Illustration of point grouping using convex hull.

4.2.2. Algorithmic Analysis of Two-Dimensional Time Series

The research on analysis of two-dimensional time series (i.e., trajectory mining) considers analysis and description of important properties in trajectory data. Of primary concern are methods to define appropriate similarity

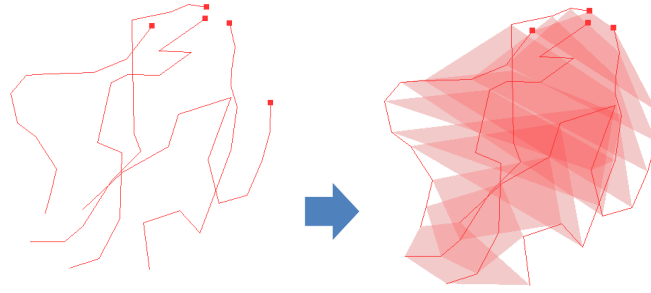


Figure 4.5.: Illustration of group movements. Left: Trajectories of individual points in the group. Right: Trajectories of individual points in the group together with the trace of the convex hulls around the group in each time point.

functions to query, compare, and cluster trajectories [NP06, PKM*07, COO05], and to support the detection of interesting patterns [PBKA08b].

In this section, we concentrate on clustering methods used for analysis of two-dimensional time dependent data.

Recently, clustering of trajectory data has received considerable attention in applications in geo-spatial and related research areas. However, finding an appropriate clustering method for trajectory data is a challenging task, as stated by Nanni et al. [NP06]: “*Spatio-temporal trajectory data introduce new dimensions and, correspondingly, novel issues in performing the clustering task. Clustering moving object trajectories, for example, requires finding out both a proper spatial granularity level and significant temporal sub-domains. Moreover, it is not obvious to identify the most promising approach to the clustering task among the many in the literature of data mining and statistics research; neither it is obvious to choose among the various options to represent a trajectory of a moving object and to formalize the notion of (dis)similarity (or distance) among trajectories.*”.

In clustering, the definition of similarity (or distance) between entities plays an important role. The clustering of trajectories mainly uses either euclidean distance or transformation of trajectories into feature vectors thereby defining distance between trajectories as distances between feature vectors. For more detailed discussion of measuring similarity between trajectories, we refer to Section 4.6.2.

A variety of clustering methods for trajectories have been introduced by now. We overview them below. They can be divided by the type of clustering used, according to the usage of trajectory fractions and according to the consideration of grouping of trajectories. We first discuss various clustering methods for full trajectories without grouping, then present a method for clustering trajectory fragments and finally mention a method for discovering moving clusters (groups of objects).

The clustering methods for trajectories include probabilistic clustering based on mixture of regression models, k-means clustering, hierarchic agglomerative clustering, density based clustering and SOM-based clustering.

Probabilistic clustering using mixture of regression models was proposed by Gaffney et al. [GS99]. They apply unsupervised learning using EM algorithm. This approach was extended in later works [CGS00, Gaf03, Gaf04, CGMS03] using random effects regression mixtures. Alternatively in this context, a mixture of hidden Markov models was used by [Smy97, ASKP03]. These rather sophisticated methods need an assumption of an underlying model of the data.

Direct distance based clustering using *k-means and hierarchical agglomerative clustering* was presented by Nanni [Nan02]. In the further work, Nanni and Pedreschi [NP06] propose an adaptation of a *density-based*

clustering algorithm, OPTICS in particular, to trajectory data. Their approach of temporal focusing chooses partial time intervals which are best suited for trajectory clustering. However, this approach is suitable mainly on geographic (in particular traffic route) problems, where the entities mainly follow a few main roads and many small roads leading to them. This method finds the main roads but is not very suitable for abstract data with no particular routes. This paper was extended with interactive selection of distance functions, progressive cluster refinement and interactive visualization in 2008 [RPN*08] (see Section 4.2.4 for more information).

Density-based clustering was also used to discover interesting places in trajectories by Tietbohl et al. [PBKA08a]. They concentrate on discovering stops in long trajectories of moving objects in geographic space. They do not focus on the similarity of trajectories.

Lee et al. [LH07] extend *density based clustering with trajectory partitioning* for finding clusters based on sub-trajectories. Their algorithm partitions a trajectory into a set of line segments, and afterwards groups the similar line segments into a cluster. They apply the minimum description length (MDL) principle for trajectory partitioning and density-based line-segment clustering for the grouping. The results are similar to the method of Nanni and Pedreschi [NP06].

Trajectory partition and subsequent trajectory partitioning for abstract (financial) data was presented by Schreck et al. [STFK07]. They use self-organizing map (SOM) for getting overview of the trajectory patterns, grouping them by similarity and the visualization of the results. This approach has been extended in future work (see Section 4.2.4).

An approach to *discovering moving clusters* (i.e., “sets of objects that move close to each other for a long time interval”) was presented by Kalnis et al. [KMB05]. Their algorithm performs spatial clustering in each time point and combines the results into a set of moving clusters. Two means of algorithm acceleration are proposed as well. The main difference to the above mentioned approaches is that the set of objects in a cluster may vary over time.

Please note that similarly to clustering, *trajectory aggregation* can be used for abstraction of trajectories. There exist many approaches, which are summarized in [AA08].

4.2.3. Visualization of Two-Dimensional Time Series

The visualization of time series, in general, is a broad area within information visualization. A complete overview of the literature on visualization techniques for presenting time-dependent data would exceed the scope of this chapter. Nonetheless, in Section 2.3.3, a brief overview of visualization techniques for dynamic data is provided. Additionally, surveys of systems specialized on time-series can be found in [AMM*08] or [SC00]. However, these surveys focus on the representation of one-dimensional time series.

In the following, we concentrate on techniques specialized on the visualization of two-dimensional time dependent data. The visualization of this type of data in a dynamic display enhances the static visualization of two dimensional points with time dimension. In the following, we consider both the static and dynamic case. We first discuss techniques disregarding grouping of entities and then those techniques which include the grouping information.

4.2.3.1. Visualization of Two-Dimensional Time Series Disregarding Grouping of Entities

The visualization of two-dimensional data in the static case usually employs scatterplots. The techniques for dynamic data therefore often enhance scatterplots with a visualization of the time dimension of the data. They employ animation of points in 2D [CK03, AAG00] or visualization of trajectories in 2D [NFA01] and 3D [Kra03]. Some systems combine both approaches [Gap, TK07]. An evaluation of the techniques was presented

in [RFF*08] and in [TK07]. The results show that animation is more suitable for presentation/overview examination, while trajectories are more suitable for the detailed analysis of the data.

In the geographic domain, also aggregated views (such as spatio-temporal histograms), T-T (time-time plots) have been introduced. An overview of geo-based visualization techniques is provided in [AAK*08]. Additionally, Willems et al. [WvdWvW09] presented a technique based on density fields displayed as colored height maps. It offers exploration of vessel movements and their speed variations.

4.2.3.2. Visualization of Two-Dimensional Time Series Including Grouping of Entities

Two dimensional data (i.e. 2D points) can be grouped according to a selected criteria into so called 2D point clouds. Such point clouds, in the static case (or in each time point), may be represented by solid shapes, using various geometric constructs (so called “*hulls*”) or using distance fields. In [SP07], the comparison of various hulls was shown. The hull types include minimum bounding discs, boxes, and convex hulls. In [SSZW08], an algorithm for the construction of compact, enclosing shapes was presented. Recently, so called “bubble sets”, continuous isocontours connecting group members were presented [CPC09]. *Distance fields* allow for the representation of point sets by smooth formation of visual areas, using appropriate transfer functions [KTSZ08]. On the efficiency side, the visualization of massive point cloud data sets may be accelerated by appropriate data structures as presented in [HE03]. The common challenge of these approaches is a compact representation of the cloud revealing the shape and distribution of the points while avoiding the overlapping of the clouds.

In analogy to the visualization of 2D dynamic points described above, visualization of 2D point clouds over time can also employ animation, trajectories of clouds or combination of both. The hulls around trajectories are however suitable mainly for points that move closely together and disregard inner distribution of the entities.

For spatio-temporal analysis of multiple entities, in the geographic domain, several aggregation-based approaches have been introduced which are surveyed in [AAK*08]. These include aggregating movement data into a surface by computing the total number of person-minutes spent in each cell of a regular grid, transition matrix counting number of entities moving between each pair of locations, and discrete or continuous flow maps. These approaches however suffer either from spatial context, or overplotting issues.

4.2.4. Visual Analysis of Two-Dimensional Time Series

Related work to the visual analysis of two-dimensional time series includes approaches which combine algorithmic analysis of the data with interactive visual exploration of the data. In analogy to the previous, we first introduce approaches which disregard groupings of entities and then concentrate on those which include groupings in their method.

4.2.4.1. Visual Analysis of Two-Dimensional Time Series Disregarding Grouping of Entities

The visual analysis of trajectories is a relatively new topic dealt with mainly in the geographic applications. The study of Andrienko et al. [AAK*08] presents an overview of Visual Analytics techniques for the detection of patterns in movement data. They discuss both the algorithmic and visualization techniques for individual and group movements while stating challenges for future research in the geographic spatio-temporal research area. In the following, we present selected recent studies on visual analysis of movement patterns presented.

Wren et al. [CZQ*08] visually analyze facility monitoring data. The dataset includes sensor information on movements of people and camera pictures without direct entity identification (i.e., they are not able to identify persons passing sensors). They offer interactive visualization of movements, the sensor occupancy distribution

and path queries. From the algorithmic analysis point of view, a new track recovery algorithm is proposed. However, given that we consider trajectories of identified entities, the latter part of their approach is not relevant to the work presented in this thesis.

Spatio-temporal visual analysis of individual movements are presented by Andrienko et al. [AA08] who propose *spatio-temporal aggregation* for visual analysis of trajectories. The spatial aggregation algorithm is combined with interactive visualization of movements showing main data flows.

Recently, Andrienko et al. [AA07] presented a system which supports visual analysis of car movements using a combination of interactive visualization and clustering and aggregation of routes. The visualization is geo-based in 2D and 3D with options to select trips and time periods. The clustering of trips according to, for example, start and end points employs the OPTICS algorithm. Clustering results can be interactively visually explored. This work has been extended using progressive density-based clustering with interactive selection of distance functions by Rinzivillo et al. [RPN*08]. For example, they use clustering of trajectories by starting and finishing points first and then refine the clusters according to route parameters. They provide an interactive exploration of the results using state of the art geographic visualization techniques. This is an extension of a previous paper [NP06] with an interactive selection of parameters and a visualization of results (see Subsection 4.2.2). This study is mostly similar to our work.

Visual analysis of trajectories using SOM clustering and interactive exploration of the results was presented earlier in 2007 also by Schreck et al. [STFK07]. This study was extended with interactive visual monitoring of the clustering process and steering of initialization in 2008 [SBTK08].

4.2.4.2. Visual Analysis of Two-Dimensional Time Series Including Grouping of Entities

The visualization of *static one-class point data* extended with statistical analysis of the data was explored in [WAG05] and [WAG06]. The statistic indicators of point cloud shape and point distribution are used for proposing interesting two dimensional projections of the originally multi-dimensional data for further visual inspection. This approach is however limited to static data without entity groupings (only one class of data).

In the *dynamic* case, the ESDA toolkit [BK04] offers the possibility to visualize hulls around trajectories and calculate and visualize the central tendency and dispersion of the group movement. This approach does not include further analysis of the data or other visual abstractions and is mainly suitable for entities moving together.

4.2.5. Summary

The algorithmic analysis of two-dimensional time dependent data is mainly concerned with the similarity and the clustering of trajectories. The similarity measures used are mainly based on direct trajectory or feature vector representation. The clustering algorithms used include, for example, k-means, density-based and probabilistic approaches. In 2007, we presented a system for visual representation of clusters of trajectories using SOM. Additionally, there are approaches for trajectory aggregation and clustering of sub-trajectories.

The visualization approaches for two-dimensional dynamic data include 2D and 3D animation and trajectory techniques based on scatterplot framework.

The visual analysis of two-dimensional time-dependent data without grouping information combines the above-mentioned algorithmic analysis with interactive visualizations for exploration of the data space.

The approaches for groups of data in the dynamic case are rare. In the static case, statistic analysis of a point cloud has been used for finding interesting views on the data. In the dynamic case, visual abstractions (e.g., hulls, mid-points) are used for visualization suitable for data exploration.

In summary, the tight integration of algorithmic analysis with interactive visualization steered by the user for analysis of trajectories both in individual and group case has not been extensively explored. Individual fields include a variety of methods which are applied mainly separately. Moreover, there are only few approaches for visual analysis of this type of data.

4.3. New Approaches to Visual Analysis of Two-Dimensional Time-Dependent Data

In order to address the three analytical tasks in visual analysis of two-dimensional time dependent data described in the introduction of the chapter, we have developed three interrelated approaches (see Figure 4.6). For the exploration of the dynamics of individual items, we provide appropriate interactive visualization tools based on the scatterplot framework that are supported by the results of our perception study on animated visualizations of dynamic two-dimensional data. For the analysis of a groups's dynamics we combine interactive visualization of the data extending the visualization means from the first task to the visualization of group dynamics. Owing to possible strong overplotting in the data visualization we provide interactive tools for feature extraction for the time-dependent data, which allow for the identification of interesting views on (interesting events in) the data. The analysis of patterns in the two-dimensional dynamic data is supported by trajectory clustering combined with interactive visualization and steering of the clustering process and clustering results together with an assessment of clustering quality.

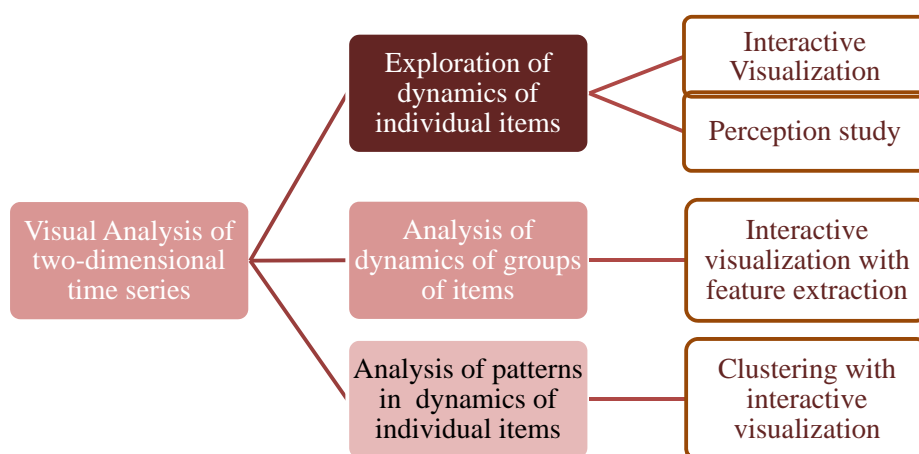


Figure 4.6.: The general overview of the proposed approach to the visual analysis of two-dimensional time dependent data supporting three selected analytical tasks.

4.3.1. Approach to Interactive Visual Exploration of Two-Dimensional Time Dependent Data

In our approach for the exploratory visual analysis of the data, we rely on effective interactive data visualization. The visualization techniques for two-dimensional time series are usually based on scatterplots and employ *animation and trajectory* visualization in 2D and 3D. We employ both animation and trajectory techniques while concentrating on 2D display of the data which is more familiar to the analysts than the 3D presentation (see Figure 4.7 left). We employ in both visualizations interactive functions to support the analysis. Interactive functions include, in addition to view transformations, filtering and thresholding (dynamic queries).

Animation is well-suited for visualization of a broad class of time-dependent data, wherein “most promising uses of animation seem to convey real-time changes and reorientations in time and space” [TMB02]. For example, in financial analysis, animation can help to analyze development of financial indicators over time.

When using animation, *perception* plays an important role for the efficiency of the visualization. In order to study the effects of animation setting on the awareness of dynamic changes in a dataset, we conduct a *perception study*. Our study is inspired by the main issues of interest of financial analysts – revealing similarity/dissimilarity in the development of financial indicators. In instrumental terms, the similarity is deemed to be detected due to a coherent motion of a glyph subset. The task appears, however, non-trivial because of a large number of data items (i.e. stocks) along with idiosyncratic movement of visualized items.

We wish to find out whether there is an effect of animation velocity and the size of direction change on the detection of the direction change by the user (see Figure 4.7 right). In a laboratory experiment, we employ a simplified alteration of an animated scatterplot of time-dependent financial data used for exploration of the two-dimensional time dependent data space while changing data and visualization parameters. The results will provide guidelines for setting-up of animation parameters in the visualization.

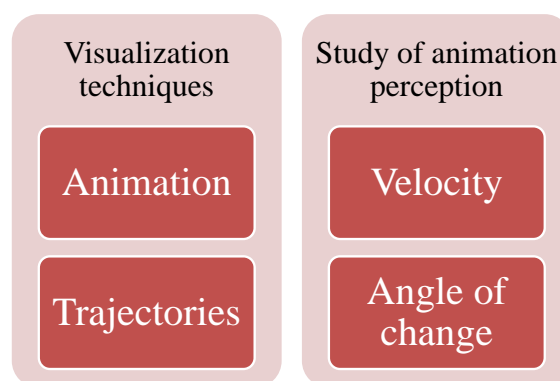


Figure 4.7.: Approach to visual exploration of two-dimensional time-dependent data. Left: The two main visualization techniques used: point animation and display of trajectories. Right: The study of animation perception with the animation parameters in focus.

Trajectory shows the path of a point in two-dimensional space (scatterplot). While animation is good for understanding general movements in the data (shown in our study in [TVK08]), it is however difficult to follow

exact movements of individual items over longer time periods. Therefore, alternatively, we use trajectories for showing the exact paths of the items.

4.3.2. Approach to Visual Analysis of Two-Dimensional Time Dependent with Grouping of Entities

In this section, we concentrate on the *visual analysis of groups of time-dependent two-dimensional data entities*⁵. We assume a constant grouping of entities according to predefined grouping criteria.

In our approach, we combine algorithmic data analysis with interactive data visualization of dynamic point clouds (see Figure 4.8). In many application areas, *point cloud visualization* can be an effective tool for data analysis. The visualization of point cloud data for their exploration has been used in geography [PSKN06,HK98], microarray data [CK03] or database exploration [PKJ*07]. In our work, various data visualizations using hulls and path traces are offered for the exploration of the data set. On the analytical side, we extend the approach of Wilkinson et al. [WAG05] for *algorithmic analysis* of single group static two-dimensional data into multi-group time-dependent data.

When algorithmically analyzing time-dependent 2D point clouds, several aspects need to be taken into consideration. In particular, we consider time-dependent features (i.e., indicators, characteristics) of

- *points within groups* (e.g., velocity, position within group, etc.),
- *each group* (e.g., changes in point density, point cloud movements, etc.),
- *multiple groups* (e.g., overlap, distance, etc.).

In the case of a *single entity in a group*, the movement and the position of the point within the group is relevant (also with respect to the group movement). For example, it is relevant whether the entity is an outlier in the group, lies in the center or at the border of the group. For the dynamics of the entity, the change of these indicators, such as the value and change of movement directions, and the length of movement is very interesting for the analysis.

For a *group of entities*, the distribution of entities within the group, the center, size and shape of the point cloud are the indicators of main interest. For time-dependent point clouds, the changes in these indicators as well as changes in the absolute position of the cloud are relevant. The size and shape of the cloud can be represented by enclosing hulls (alpha, convex, butterfly, circle, etc.) which can be visualized and for which we can calculate various descriptors (area, convexity, direction, etc.). Mid-points of entity groups and their movement serve as an additional indicator of group position and its change over time.

Multiple groups can be characterized by their relative position to each other (e.g., number of overlapping points, the relative overlapping area, number of overlapping point clouds) and by their dynamics (co-movements) characterized by the dynamics of the mid point.

The proposed features are calculated and visualized (see Figure 4.8). These time-dependent characteristics are used for the determination of interesting data views. This is especially useful when analyzing large amounts of data (with respect to either time-dimension or number of objects). The features can indicate time-periods and data elements which are deemed useful for further detailed visual inspection (see Figure 4.9 for an illustration). For the details on the indicators used in the analysis, we refer to Section 4.6.2.

⁵In the further text also referred to as time-dependent/dynamic groups, time-dependent/dynamic point clouds, or time-dependent/dynamic data classes

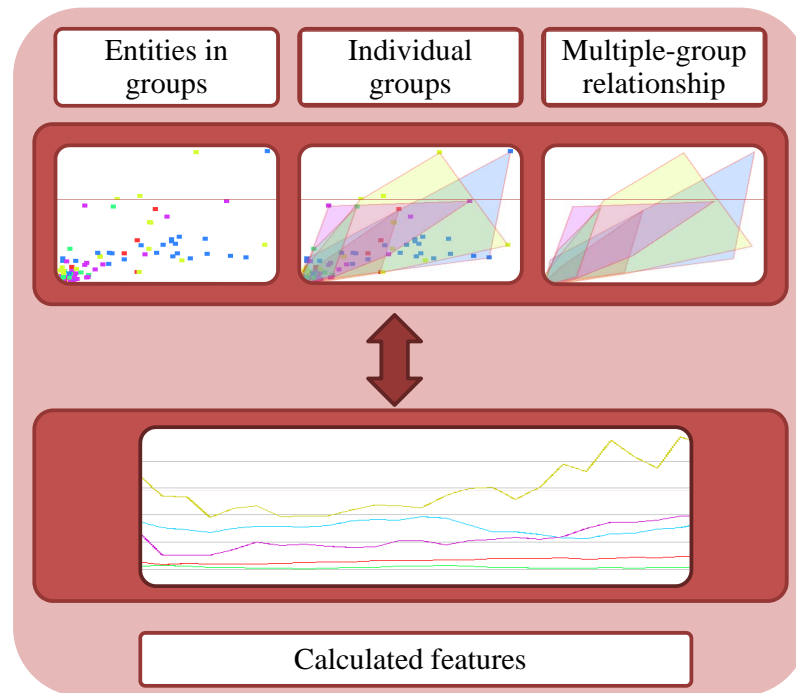


Figure 4.8.: Approach to visual analysis of groups of two-dimensional time-dependent entities using feature monitoring. The data is visualized and algorithmically analyzed on three levels: the development of individual entities within groups, of individual groups and of relationships between individual groups over time. The extracted features are visualized and explored. These results reveal interesting parts of the data set, which can be visually analyzed in more detail on demand.

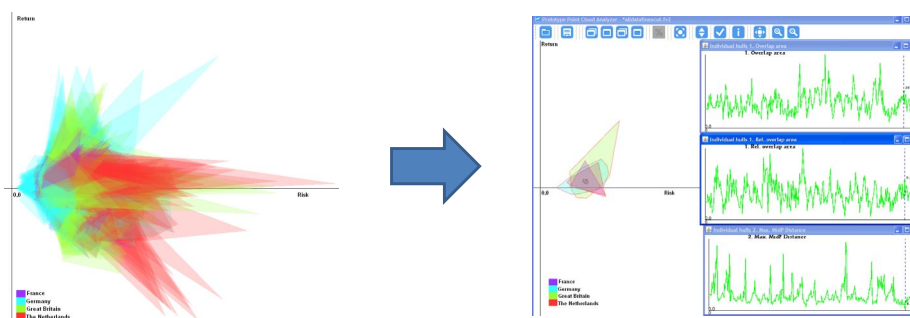


Figure 4.9.: Illustration of visual analysis of groups of two-dimensional time dependent entities. Left: State-of-the art visualization of the grouped data using convex hull trace visualization showing an overcrowded display that is difficult to explore. Right: The new feature-based visual analysis revealing interesting views on the data. These interesting views are in the focus.

4.3.3. Approach to Visual Analysis of Two-Dimensional Time Dependent Data Using SOM Clustering

This section presents the part of the system suitable for visual analysis of large amounts of two dimensional time series (i.e., trajectories), in particular, with regard to the patterns in trajectories. In Section 4.3.1 an visual exploratory approach for analysis of two-dimensional dynamic data was presented. However, with increasing number of time steps or number of entities, strong overplotting and limited perception capabilities reduce the capacity of similarity and pattern detection in the data. We propose a clustering algorithm for examination of a large number of trajectories by abstracting to a limited number of prototypes describing groups of similar trajectories and thereby providing an overview over the whole data set (see Figure 4.10). In this respect, we follow Keim’s Visual Analytics mantra “*Analyse First – Show the Important – Zoom, Filter and Analyze Further – Details on Demand*” [KKMT06]. The classic clustering steps as also used in general clustering framework VISTA [CL03], include data pre-processing, initialization, clustering, quality assessment and post processing. In addition, also steering of the clustering learning part is provided. This approach is in analogy to the graph analysis presented in the Section 3.3.3.

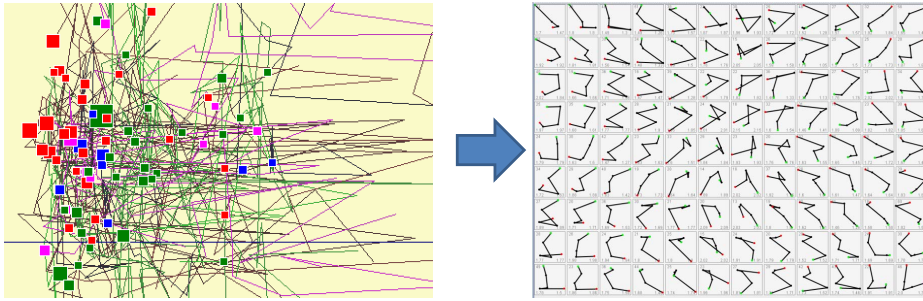


Figure 4.10.: Illustration of the new visual clustering of trajectories using SOM. Left: State-of-the-art trajectory visualization of the data showing and overcrowded display that is difficult to analyze. Right: The display of the result of the new approach using SOM-based clustering showing an overview of the trajectory patterns occurring in the data set.

Clustering of trajectory data requires the definition of an appropriate similarity concept. In principle, many different similarity concepts are possible; based directly on the geometry of the trajectories, or on derived numeric, structural, or symbolic features (see Section 4.6.2 for details). Furthermore, the space of trajectory similarity notions is broadened by data preprocessing choices such as normalization steps applied, and the level of detail considered. In our approach, we use alternatively direct geometric representation and abstract trajecory features with variable normalization possibilities.

There are many clustering techniques available (see Section 2.5.2.1 for an overview). In our approach, similarly to the graph clustering introduced in Chapter 3, we use self-organizing maps as a clustering algorithm because it supports well large datasets and provides reasonable results with direct visual output.

In our work, a flexible system for trajectory clustering is proposed. It consists of several algorithmic and visualization steps which can be interactively steered (see Figure 4.11). In the standard approach, the SOM clustering is produced by an unsupervised training process which ends once a fixed number of iterations has elapsed or the quantization error meets a predefined threshold [Koh01]. This approach was used in graph clustering in Section 3.6 and can be used also for trajectory clustering. However, in an advanced mode, we aim to produce

SOM cluster results that are both good with respect to quantization error, and at the same time reflect user- or application-desired prototype patterns and layout criteria. We therefore extend the unsupervised training process (a) by on-line visualization, and (b) by control functionality. Visualization of on-line training and optional user intervention are coupled. At any time during the training, the user is able to pause the training, update training parameters, and resume the training. We point out that this extension is not required in every data analysis scenario and is mainly an expert feature.

The clustering-based visual trajectory analysis system (see Figure 4.11) is similar to the clustering system for visual graph analysis. It starts with fragmentation (i.e., partitioning) of trajectories into smaller parts.⁶ For calculation of similarities, each extracted trajectory fragment is described by a set of features creating a feature vector data set. In our case, this feature set consists of geometry and/or abstract features. The partitioning and feature extraction results can be visually explored using various views (e.g., using multivariate visualization techniques such as parallel coordinates). In the following, the features can be adjusted (weighted/selected) using a user interface. The selected feature set is used as input for calculating similarity during clustering. After choosing the SOM clustering parameters and, if needed, setting up initial SOM layout, interactive SOM clustering is performed.

During the learning process, the intermediate results are shown and assessment of the clustering quality is performed and displayed (see Sections 4.6.4, 4.6.5 and 4.6.6). Within the learning process, the users may pause and refine SOM clustering parameters on demand. The intermediate and final results can be further explored using interactive views. The results and initial parameters can be stored for reproducing and comparison of results. The feedback loop allows to change parameters and rewind the process on demand.

Finally, intermediate and final results can be interactively explored. For the interactive visualization, many of the interactive views presented in graph clustering (see Section 3.6.5) can be applied also to trajectory clustering visualization. However, owing to the additional time dimension and possible use of direct geometric representation of the data for the display, additional possibilities for visual exploration of the results emerge. For example, we can offer time filtering of the results, entity-based filtering or sequential views on the data. When applying geometric representation of trajectories as features, we can use the cluster center features as direct geometry for visualization, i.e., there is no need for abstract visualization methods for feature vectors such as parallel coordinates.

⁶In case it is suitable for the analysis and is not already provided as input.

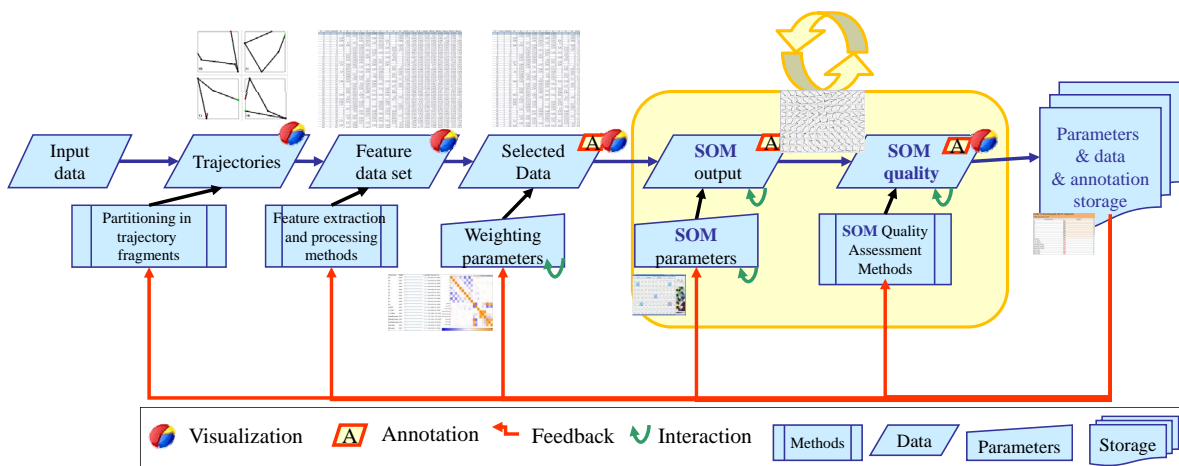


Figure 4.11.: The process of visual analysis of trajectories. The system supports trajectory fragmentation, trajectory feature extraction, SOM clustering and interactive visualization and assessment of SOM clustering process (highlighted in yellow) and its results.

4.4. Interactive Visual Exploration of Two-Dimensional Time-Dependent Data

4.4.1. Introduction

In this section, we first present our tools for interactive visual exploration of two-dimensional time series using animation and trajectories in a scatterplot framework combined with interactive view and data change functions. We then describe the set-up and the results of the study on perception of direction changes in the visualization of data dynamics in an animated scatterplot. The results of the study help the design of animated visualization systems.

4.4.2. Interactive Visualization

4.4.2.1. Visualization

The initial visualization shows four dimensions of the whole data set at the starting time period in a scatterplot (see Figure 4.12). Data are presented as rectangles, where x position, y position, color and size each encode one dimension of the data. Please note that node size and color are optional parameters used on demand. For encoding the size the glyph area instead of side size of the rectangle is used, because humans perceive size as area. The axis minimum and maximum are determined by the data over the whole time period so that the glyphs do not move outside the plot area during animation, however it can be adjusted by the analyst if necessary.

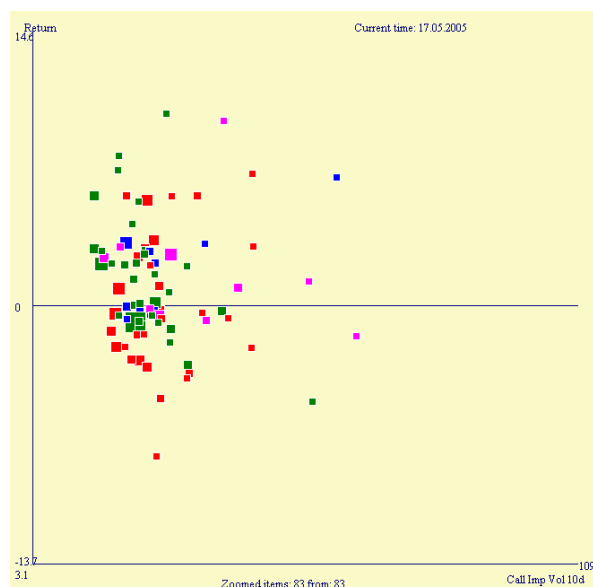


Figure 4.12.: Visualization of the time data in a scatterplot showing a snapshot of the data in a single time point. The x-axis and y-axis encode the two dimensions of the data. Additional dimensions, if provided, are displayed using color and glyph size. Animation is used to represent the time-dimension of the data.

With an increasing number of cross-sectional entities, the overlap of glyphs and labels will pose problems for visualizing the data. In order to separate the glyphs, we firstly have chosen to use a different color for the glyph's borders from its "filling" color and secondly to position glyphs on the back/foreground of the screen according to their size. The border color highlights contours of the glyphs and back/foreground positions controls for overdrawing smaller glyphs with bigger glyphs.

Animation Animation shows the time dependent changes in the data. The linear movement of the glyphs represents the data change between two subsequent points in time. The animation steps are smooth in order to support the accuracy of decision making [Gon96]. There are different options for the choice of interpolation technique which can be used for animating scatter plot glyphs. We have employed a simple linear movement between two subsequent data points, because this motion is intuitive and easy to follow. In linear interpolation, the speed of motion is proportional to the rate of change in the data between two subsequent time points, as advised by [AWS05]. When using animation, the animation speed and possibility of stopping and replay is essential in order to better concentrate on changes in glyph positions during the play [TMB02]. Therefore we integrated these functions as well.

Trajectories The animation of the glyphs is a powerful tool for showing data changes. However, it disregards the history of past positions of the data points. Therefore, the system provides users with the option to draw trajectory paths of the items during animation. This feature is particularly important for users interested in analyzing and comparing the history of data items in more detail, when viewing a "static picture" of the data (see Figure 4.13). As for longer time periods, the trajectories suffer from strong overplotting, fading out effects for a chosen period (e.g. 5 days) is used as well.

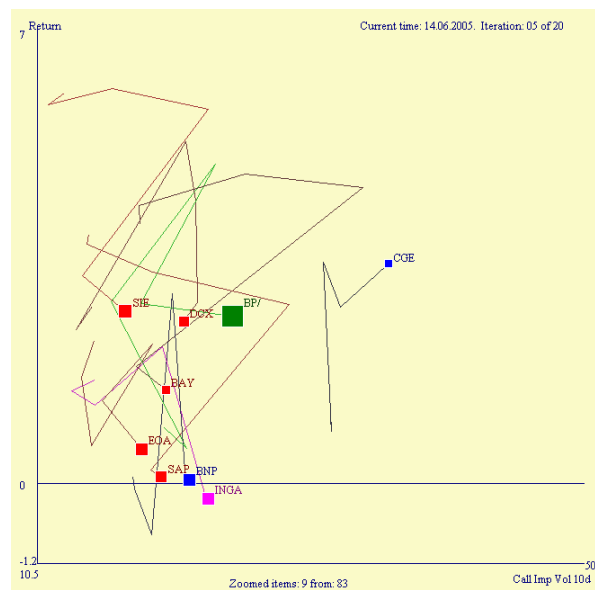


Figure 4.13.: The visualization of the data using trajectories.

4.4.2.2. Interaction

As analysts need to examine large datasets, interaction features are required in order to provide users with the possibility to concentrate the analysis on interesting data. The main interaction functions used include filtering, highlighting, panning and zooming.

There are several filtering and highlighting options according to data values. For example, items that satisfy a user defined threshold on the dimensions represented by the x- and y-axis or the rectangle size are possible. For example in the financial domain, the filtering/highlighting of stocks according to size allows to compare the dynamics of different stock groups – big companies versus small companies. The x and y thresholds show whether the companies satisfying the risk and return criteria in one time period keep on satisfying these limits over time (see Figure 4.14).

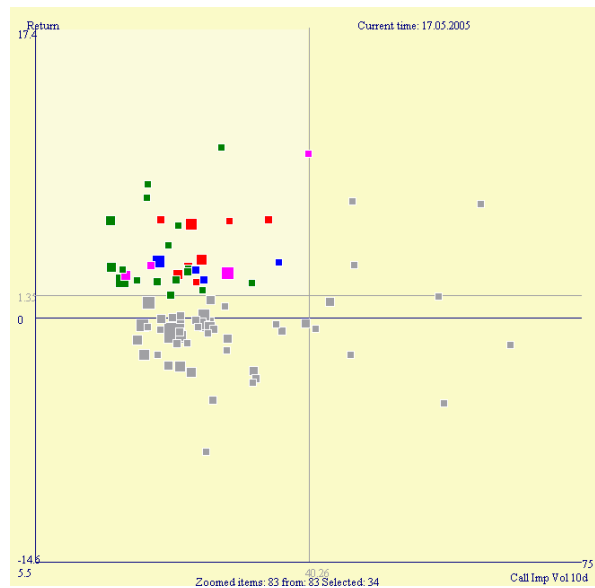


Figure 4.14.: An example of interactive data exploration using filtering/highlighting by thresholds on x and y dimension. The filtering/highlighting is used for focusing on a part of the data set of analytical interest. In this example, the stocks with high return and low volatility (those with good performance indicators) are selected for detailed analysis. The observation of the data over time reveals their stability in the high-return-low-volatility region.

4.4.3. Perception Study for Visualization of Two-Dimensional Time Dependent Data Using Animation

In our study, we investigate how certain features of an animated display affect detecting of changes in dynamic data. In particular, the interest is in the parameters of a coherent motion of items that affect detection of an item subset of interest, while consistently varying (i) the velocity of item motion and (ii) direction difference of the subset. The rate of correct responses and reaction times serve as performance measures. In addition, possible gender differences under the same conditions are tested.

We first present a brief survey of the main issues of motion perception in the context of data visualization employing animation. We then describe the set-up of the experiment and then the results. Finally, we discuss the results.

4.4.3.1. Perception of Motion: Issues Related to Visualization

The review of related work focuses on perceptual phenomena related to detection of change in motion direction contingent on motion velocity and the angle of direction change, the two variables of interest in our experiment.

Design Guidelines for Perception of Motion in Visualization Although motion is used in numerous visualization techniques (e.g., [SH06], [VM04], [VMMG04] and [HH05]), “*the detailed knowledge needed to construct perceptual guidelines on the use of motion in visualization has not been documented*” [TMB02]. One of the most broadly set guidelines for designing animated visualization are in [AWS05]. Since our work focuses on direction discrimination and motion velocity, we refer to the following guidelines:

- According to the Gestalt principle of the “*common fate*”, objects moving together are assumed to be related.
- Motion perception is poor at low velocities (lower than $4^\circ/\text{s}$) and at high velocities (beyond $70^\circ/\text{s}$).⁷
- Two motion directions should differ by at least 1° to be perceived as such. However, direction discrimination improves as exposure duration lengthens and as the size of a stimulus increases [WS92].

In addition, it was found in [NQ99] and [GBH98] that the ability to identify direction differences changed considerably by varying the axis-of-motion: direction discrimination thresholds are significantly higher for motion in oblique directions compared to that in cardinal (horizontal and vertical) directions (the oblique effect of motion). Huber and Healey [HH05] provide, in particular, certain measures of detecting changes in direction and velocity of moving objects, the aspects critical in the present context of distinguishing difference in motion of glyphs. These authors show that for a change to be detected, the velocity should deviate by at least $0.43^\circ/\text{s}$ and the angle of direction change by at least 20° .

Discrimination of Speed and Direction Change of Motion Hohnsbein and Mateeff [HM98] suggest that changes in the direction of motion may be detected based on the perceived velocity change the items produce. De Bruyn and Orban [DO88] report that the threshold of human direction discrimination is described by a U-shaped function dependent on the stimulus velocity: the thresholds decline up to the velocity of $4^\circ/\text{s}$ and remain constant up to $128^\circ/\text{s}$; for velocities faster than the latter, thresholds rise again.

Ball and Sekuler [BS87] and De Bruyn and Orban [DO88] found an improvement in direction discrimination with increasing velocity up to a critical velocity, with the latter comparable in both studies, 8° - $10^\circ/\text{s}$ and 4° - $8^\circ/\text{s}$, respectively.

⁷◦ refers to visual angle of the person sitting in front of the display.

Perception of Motion of a Group of Objects In the analysis, there is seldom the need of detecting motion of an object appearing in isolation. Detection of a group of objects' motion relative to another set of objects is the main issue of the analysis and therefore also of this study. As Sekuler et al. [SWB02] report, humans are able to detect just 5% of coherently moving dots in a pool of randomly moving dots. Absolute thresholds, though, were found to vary with the display size, item density, and exposure duration, but not with the direction of motion.

4.4.3.2. Experiment Method

Twenty subjects (mainly university students) participated in a laboratory experiment. The subjects were 20 to 30 years old (mean 25.7 ± 3.6 years). All subjects had normal or corrected-to-normal vision, in the latter case wearing their habitual glasses. All were color-normal and right-handed. The experiment was carried out individually. Each subject was sitting in front of a monitor at a distance of 65 cm. Ambient daylight room illumination (3000 lux) was held constant.

Prior to the experiment, each subject underwent a training phase to familiarize her/him with the display and the task.

For stimulus presentation, a white square-shaped window was employed, with the area confined to 700 x 560 pixels ($17.4^\circ \times 17.4^\circ$ of visual angle), on a dark background. The white background had CIE chromaticity coordinates $x=0.33$, $y=0.33$ (7100 K) and luminance $L_{bg} = 160 \text{ cd/m}^2$, as measured by 'Colorimeter HCT-99' (Fa. Gigahertz-Optik). Inside the window, a pool of randomly distributed glyphs (N=30) was presented. The glyphs were equally-sized green squares (11 x 11 pixels, or 0.28°), with the CIE coordinates $x=0.29$, $y=0.356$ (RGB: 0,128,0) and luminance $L_{sq} = 118 \text{ cd/m}^2$. The rectangles had white contours in order to overcome possible smearing items due to overlapping (see Figure 4.15).

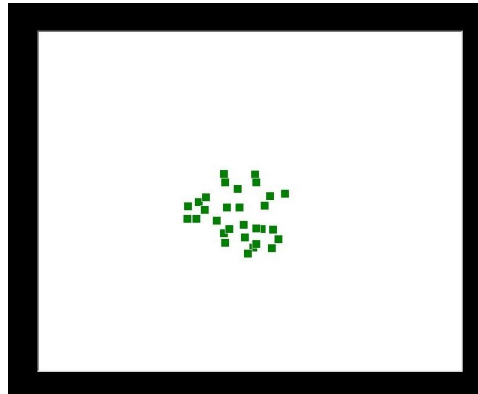


Figure 4.15.: Display of a pool of 30 random glyphs at the start of each trial as seen by the experiment participants. The white rectangle determines the region of possible glyph positions.

Upon onset of a trial, the pool of randomly positioned items started moving smoothly and linearly with a constant velocity. Between blocks of trials the item velocity was varied, being either 'fast' ($VF = 0.25^\circ/\text{s}$) or 'slow' ($VS = 0.12^\circ/\text{s}$). In addition, the start direction of pool motion (D_{pool}) was varied taking pseudo-randomly one of eight possible directions. After 20 time steps (3.75 s for 'fast', 7.5 s for 'slow'), a random subset of items ($M=10$) changed abruptly their motion direction (D_{sub}), while moving as a coherent group in another direction, with the same velocity. The angle of the subset direction difference relatively to the pool was assigned pseudo-randomly from a set of angles: $\{\pm 45^\circ, \pm 90^\circ, \pm 135^\circ, 180^\circ\}$ (see Figure 4.16 for an illustration). The choice of

the eight motion directions and the respective motion direction changes as well as the two paces of the set motion set up was motivated by parameters of animated data visualization (see Section 4.4.2). Such parameters, in their turn, reflect real-world analytic tasks in financial data analysis. It is worth noting that analysts, who watch the two stock indicators, risk and volatility, are primarily interested in coarse dynamic measures of stock excursions – as represented here by major axes of change in glyph motion direction. For example, a glyph movement in the right-up direction implies an increase in return and, as well, in volatility of the asset; in contrast, a straight down movement of a glyph indicates decrease in return, with unchanged volatility. In the former case, in accordance with an economic theory, the increase in return is bound with higher risk; in the latter case, the “straight-down” movement indicates the change in the risk-return profile of the asset.

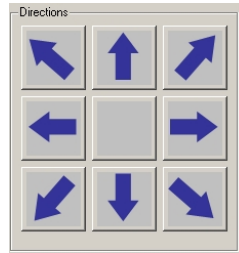


Figure 4.16.: Illustration of directions of moving glyphs (“direction wind rose”) employed in the experiment; direction differences equal 45° .

The subjects were instructed to observe the pool of moving items and, upon detecting the direction change in motion of certain items, to mouse-click those items as quickly as possible using her/his right index-finger. To indicate the clicked items, the color of the marked item changed immediately from green to black (see Figure 4.17 for illustration).

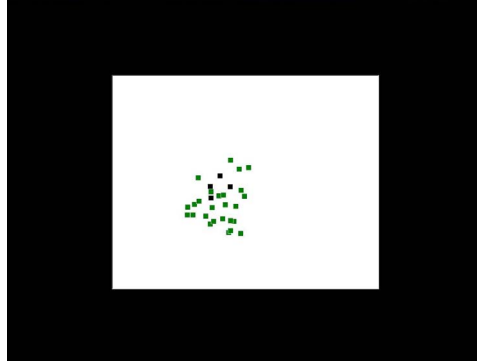


Figure 4.17.: An example of a display snapshot following a participant’s detection of the items moving in a deviant direction. The detected times are highlighted in black color.

The time allocated to participants’ responses was limited to 18.8 s for the ‘fast’ velocity and to 22.5 s for the ‘slow’. The trial terminated after the allocated time has expired, even when not all items with differential direction had been detected. The next trial was initiated after an interstimulus interval (ISI), which varied between 3 s and 7 s (to preclude participant’s time learning, which might confound a true effect of the direction-change detection). In addition, after every seven trials a longer interval of 40 s was introduced (allowing participants a brief break in

the rather demanding task); during it a white frame was presented. Between each of the trial blocks (see below) there was a 2-min break.

At each velocity, permutations (56) of the starting pool direction ($D_{pool} = 8$) and the direction differences of the subset ($(D_{pool} - D_{sub}) = 7$) were presented in a pseudo-random order, split in two blocks with 28 trials each. The subjects participated in two sessions, each including four blocks of trials. Within a block, either 'slow' or 'fast' motion was kept constant. The velocity alternated between blocks starting with the 'slow' one. In total, the experiment consisted of 224 trials (= 2 velocities x 56 direction differences x 2 sessions) and took about one hour for each subject.

Subjects' responses were registered by the computer. Reaction times (RTs) were measured up to 1 ms. In the further data analysis, RTs of the first mouse click were considered. The rate of correct responses, i.e. the number of correctly detected subset items, served as another performance measure; we also registered the rate of false alarms.

4.4.3.3. Experiment Results

For the analysis, we considered three performance measures – the rate of correct responses, rate of false alarms, and response times (RTs), defined as the time to the first correct click of a glyph as a member of the subset. Data were first subdivided according to subjects' gender and then averaged within the gender subgroup for each combination of the velocity and the angle of the direction change. Mean RTs were calculated. For the dependent variable 'rate of correct responses', the variation between subjects was found to be homogeneous. For the RTs, the variation was relatively great, across both participants and various conditions.

The primary focus of our interest was an effect of the item speed (factor *VELOCITY*). In addition, we considered an effect of varying the angle of the direction change in a glyph subset relatively to the initial pool's axis-of-motion (factor *ANGLE*, disregarding the sign) and a gender effect (factor *GENDER*).

In addition to the ANOVA test algorithm, we employed the signal-detection approach [GS66] for estimating the sensitivity parameter d' for each condition considered above. The sensitivity parameter d' is an ascertained factoring in the conditional chance of hits (correct responses to a specific stimulus) and false alarms (erroneous responses when a stimulus is not presented). With increasing sensitivity, the value of d' becomes greater, thus indicating easier stimulus detection.

As Table 4.1 indicates, the subjects were accurate in detecting the subset of items moving in a deviant direction, with the detected item number exceeding on average 9 at both employed velocity conditions.

Velocity	Angle of direction change			
	$\pm 45^\circ$	$\pm 90^\circ$	$\pm 135^\circ$	180°
'slow' (0.12°/s)	9.09	9.22	9.01	9.15
'fast' (0.25°/s)	9.70	9.52	9.62	9.53

Table 4.1.: Mean rate of correct responses over all subjects for the factors *VELOCITY* and *ANGLE*.

Oblique effects The initial data analysis has not shown any significant oblique effects – neither as a main effect [$F(3, 51) = 0.12, p = 0.66$] nor as an interaction effect [$F(3, 51) = 1.31, p = 0.28$]. In the further analysis we therefore concentrated on *ANGLE* and *VELOCITY* effects which are described in the following.

The factor VELOCITY: The factor *VELOCITY* has a significant effect on the rate of correct responses of the subjects [$F(1, 19) = 39.10, p < 0.001$] and accounts for 77.01% of variance. The effect on RTs for detecting the direction change is highly significant [$F(1, 18) = 12.07, p < 0.001$], accounting for 70.62% variance. Note, RTs are shorter at the 'fast' velocity for all direction-change steps (see Figure 4.18). The data analysis has not shown any significant oblique effects. The sensitivity measure d' (the detection of all subset items and correct rejection of the remaining items, $d' = 6.00$ as max.) indicated that direction change detection was easier at 'fast' velocity, with the d' difference from 0.24 to 0.62. For example, for the direction change of $\pm 45^\circ$, at the 'slow' condition $d' = 3.29$, whereas at the 'fast' condition $d' = 3.91$ (see Table 4.2). The same tendency is observed for the other angle parameters.

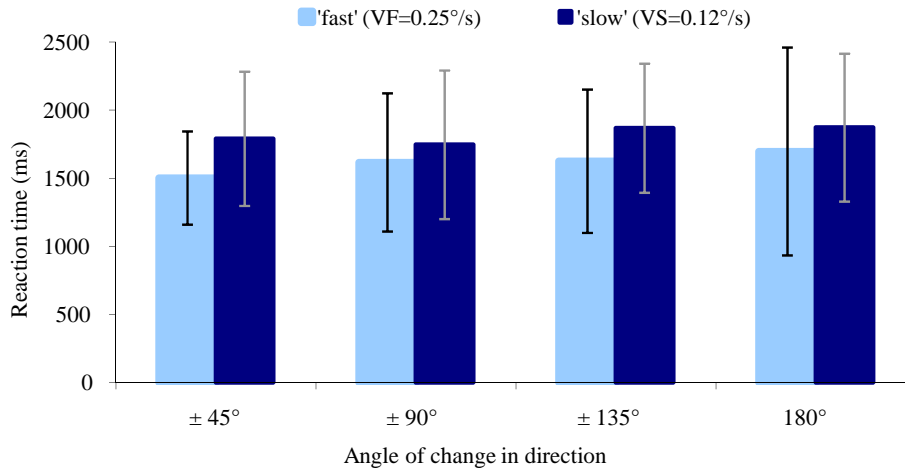


Figure 4.18.: Mean reaction times and standard deviations of the first click upon detecting change in motion direction for the two velocity conditions; parameter is the angle of direction change in the moving subset. The results indicate a significant effect of velocity and no effect of angle of change on reaction times.

The factor ANGLE We could not find any significant effects of the factor *ANGLE* on either the rate of correct responses [$F(3, 54) = 0.21, p = 0.86$], of false alarms [$F(3, 54) = 0.50, p = 0.64$], or response times [$F(3, 54) = 1.26, p = 0.30$]. Neither were there any significant interaction effects between the *VELOCITY* and *ANGLE* factors on correct responses [$F(3, 54) = 0.98, p = 0.40$], false alarms [$F(3, 54) = 0.48, p = 0.64$], or RTs [$F(3, 54) = 0.72, p = 0.50$].

Velocity	Angle of direction change			
	$\pm 45^\circ$	$\pm 90^\circ$	$\pm 135^\circ$	180°
'slow'(0.12°/s)	3.29	3.36	3.32	3.39
'fast'(0.25°/s)	3.91	3.60	3.91	3.73

Table 4.2.: Sensitivity measure d' of detecting an item subset moving in a deviant direction at the employed velocities and angles of direction change. The measure is averaged across all subjects.

4.4.3.4. Summary and Discussion

In the present study, which simulated animated-data visualization in the laboratory experiment, we were primarily interested in whether the velocity and angle of change of dynamic items affects subjects' performance, when they are asked to detect a direction change of an item subset.

We found that the ability to discriminate the direction change in a dynamic pattern is higher at the 'fast' velocity, $0.25^\circ/\text{s}$, than at the 'slow', or $0.12^\circ/\text{s}$. At the former, the participants perceive the change in motion direction of a simulated dataset faster and more accurately. The factor *VELOCITY* accounted for more than 70% of variance of both studied performance measures, the rate of correct responses and reaction times.

Our result of shorter responses at the higher velocity is in accordance with [DS82]. The results of Huber and Healey [HH05] indicate that performance measures are sensitive to direction differences below 20° (smaller than ours). De Bruyn and Orban [DO88] showed for velocities up to $4^\circ/\text{s}$ a decline of direction discrimination followed by leveling-off up to $64^\circ/\text{s}$, and an increase for higher speeds (forming a U-shaped function of speed). [HM98] found significant effects of the velocity and the angle of direction change and the interaction between them. The parameters of stimulus presentation in these studies quite differed from ours, moreover there the whole pattern changed the direction – in contrast to our design where only a subset changed direction. These differences may explain the discrepancy in the outcomes.

4.5. Visual Analysis of Two-Dimensional Time-Dependent Data with Grouping of Entities

4.5.1. Introduction

In this section, we present new methods for visual analysis of groups of entities in two-dimensional time dependent data⁸. Our implementation combines interactive visualization of two-dimensional time-dependent points in a scatterplot framework extended with visualization of point groups as whole and their distribution and center (mid-point) over time. As pure visualization may lead to strongly overcrowded displays, initial algorithmic analysis of the data helps to reveal interesting views for more detailed inspection. Therefore, the interactive visualization is combined with calculation and display of various data features on the point, group and inter-group level over time. We assume that the data groupings stay constant over the whole analyzed time period.

In the following, we concentrate on three analytical cases:

- analysis of an entity in a group over time,
- analysis of a group over time,
- analysis of multiple groups in relation to each other over time.

We first describe features we use for algorithmic analysis of the time-dependent data and then provide details on interactive visualization of the data. Finally, we discuss their combination for the analysis of the data in the three cases.

4.5.2. Time-Varying Features for Description of Groups of Two-Dimensional Time-Dependent Data Entities

In the following, we describe features used for algorithmic analysis of the time-varying scatter data. We propose new meaningful features and extend features presented in the previous works [AAPS08, AAK*08, WAG05, WAG06, vLBR09]. We describe all features used, where the new ones are depicted with “(*)”.

A large set of features is proposed, in order to cover a wide variety of possible use cases and data characteristics. For a specific use case, a selected set of relevant features needs to be considered in combination to each other in order to enable a meaningful interpretation of the data. In addition, we also provide calculation of correlations of indicators for one subject (entity, group or combination of groups) and correlations of one indicator between subjects.

The extracted features are measured at each point in creating one-dimensional time-dependent data set. They show either the current state or a state change of the data item.

For the calculation of the features, we assume data with equidistant time steps implying that velocity features are equivalent to distance (length of path). Therefore, we consider only distance in the following. For calculation of the distance, the Euclidean distance in 2D is applied. Moreover, without loss of generality, the number of points in a group is assumed to stay constant over the whole time period. Otherwise, further features or specific normalizations of the existing features can be introduced.

⁸In this thesis also referred to as time dependent groups, dynamic groups, time-dependent point clouds or dynamic point clouds.

4.5.2.1. Extended Features for Dynamic Entities in Groups

The set of features for describing the dynamics of entities in groups used in this section is an extension of trajectory features for single entities described by Andrienko et al. [AAPS08, AAK*08] (see also Subsection 4.6.2 for more details). These features describe movement of single entities without taking into consideration the point groupings. However when analyzing movements of entities in a group also relative movement and position of an entity to the group is of relevance. Therefore we extend these features with group-relevant features.

1. *Movement*: movement features describe movements of entities (following Andrienko et al. [AAPS08, AAK*08])
 - Movement length: describes the distance covered by an entity. In time dependent data, we consider
 - last step length: shows current movement speed of an entity
 - total path length: measures the sum of all movements
 - distance from start: describes the distance between current position of an entity and its position at the beginning. It shows whether the entity moved far from the start or has stayed near. When analyzing this measure over time we can see also stability of the entity position.The combination of the length of the movement and distance to the start of the movement can reveal circular or oscillating movements.
 - Direction: These measures contain current movement direction and total direction from start. The direction is measured as an angle to the X axis.
 - current direction: the direction of the last movement of the entity. It can reveal sudden turns or continuous movements in one direction.
 - total direction: the direction between current and start position shows the general tendency of the movement.
2. (*) *Position in group*: These indicators show the relative position of an entity in a group.
 - Distance to boundary: shows whether the entity's position is either more in the center or more at the border of the group. It is defined as the minimum distance to the convex hull.
 - Distance to mid point: shows whether the point is close to the center of gravity of the group (please note the difference to geometric center of group in case of inhomogeneous distribution of the entities in the group).
3. (*) *Co-movement with group*: indicates whether the entity moves in coherence with the majority of the group (approximated by the movement of the mid-point of the group).
 - Length: shows whether the speed of the point is similar to the speed of the group approximated by the speed of the mid-point.
 - Direction: shows whether the direction of the point is similar to the movement direction of the group approximated by the angle difference to the mid-point.
4. *Outlying*: indicates outliers in the groups.
 - Is outlier: indicates whether the point is an outlier in the group or not. Note that although similar to the distance to boundary, points on the boundary are not automatically outliers. The outlier definition used in this section follows Wilkinson [WAG05, WAG06] and is based on trimming of the minimum spanning tree between entities in the group.

4.5.2.2. Features for Individual Dynamic Groups

The time-dependent features for one group measure various group aspects such as shape of the group, distribution of the points in the group as well as group dynamics. The state features used follow the measures proposed for single static point cloud by Wilkinson [WAG05, WAG06].

We extend these features with further state and movement measures based on mid-point and PCA. Moreover, we introduce measures for assessing the dynamics of a group which were not regarded in Wilkinson work on static data.

1. *Group size*: the group size features include area and diameter of the point cloud. They show compactness of the entities. Please note that larger area does not automatically correlate with diameter as changes in the group shape influence both measures.
 - Area is measured as the area of convex hull around the points,
 - (*) diameter is defined as the maximum distance of points in the group.
2. *Point distribution*: measures show the homogeneity of point positions in a group. For assessing point distribution, we use measures introduced by Wilkinson [WAG05, WAG06].
 - skewness: measures the relative density of points based a ratio of quantiles of the edge lengths,
 - sparsity: indicates homogeneity of the point distribution,
 - strait: reflects “coherence in a set of points as the presence of relatively smooth paths in the minimum spanning tree” [WAG06].
3. *Group shape*: these measures try to describe the shape of the group points. We use measures (convexity, skinniness, stringiness) from Wilkinson [WAG05, WAG06] and extend them with a PCA based measure of the shape.
 - convexity: based on the proportion of the area of the alpha hull to the convex hull measures the convexity. Numbers close to 1 indicate convex shapes.
 - skinniness: indicates circular, squared or long rectangular shapes. Zero means circular shapes, and one long rectangular shapes.
 - (*) PCA relative eigenvalue: the relative proportion of first and second eigenvalue of point cloud PCA for assessing compactness of the group. A large number indicates long narrow shapes, numbers close to one indicate the opposite.
4. *Group alignment*: Not only shape but also the alignment of the points is measured. The alignment group of features is introduced here as a specific type of feature. We redefine correlation measure introduced by Wilkinson [WAG05] for assessment of group shapes from shape type into this type of feature as it is more suitable for this new alignment group of features. We extend this measure by the direction of the points based on PCA major axis direction.
 - Point correlation: the strength of the linear relationship between X and Y dimension of the data,
 - (*) major PCA direction. the direction of the major eigenvector reflecting the angle of group expansion relative to X axis.
5. *Outlying*: shows also a kind of compactness of the group. It measures the relative number of outlying points in the group (based on definition by Wilkinson [WAG05, WAG06]).
6. (*) *Movement of the group*: indicates direction and speed of group movement approximated by mid point dynamics.
 - current and total distance: the length of the mid-point current step and sum of all steps

- current and total direction: the direction of the current step of the mid-point and the direction between start and end point of the mid-point of the group.

4.5.2.3. Features for Relationship among Multiple Dynamic Groups

When analyzing multiple groups, we can either analyze each group separately and then compare their statistics or look at measures for the inter-group relations. In our work, we adopt both approaches. In the first case, features mentioned above are used for each group. Features for the second case are presented below. They include measures for relative overlap of groups and their relative distance and position. The difference between overlap and distance depends on group size (small groups close to each other may not overlap). For these indicators, we may consider multiple groups together or each two pairs separately.

1. (*) *Overlap*: shows whether groups overlap. We measure both absolute and relative overlapping area as the (relative) area of the group intersection to the united group area. When we compare two groups then also relative containing area of the smaller group in the larger group is considered. Additionally the number of groups that overlap is calculated.
2. (*) *Distance*: The distance between groups is measured as distance between their mid points. In case of more than two groups, average, minimum and maximum of the distances is calculated.
3. (*) *Relative position*: measured by direction and distance between mid points of groups. This is applicable only when comparing two groups.

Some of the features mentioned above are relevant only to the comparison of two groups. When comparing a larger number of groups, pairwise comparison of the groups can be undertaken and thereby also these features can be used.

4.5.2.4. Feature Normalization

When analyzing the features their scale may be relevant. Normalization can be done according to Wilkinson [WAG06], or min-max normalization of each feature for all entities over the whole time period can be used.

4.5.3. Interactive Visualization of Two-Dimensional Time-Dependent with Grouping of Data Entities

In the following, we concentrate on visualization and analysis of the three aspects of the dynamics of groups of entities in two-dimensional time dependent data:

- individual entities within a group
- groups

4.5.3.1. Entities in Groups

We visualize the absolute and relative⁹ position of entities (points) in a group (point cloud) by traditional scatter plots. The time development of points is displayed using trajectories (see Figure 4.19), which is also used for visual exploration of the data (see Section 4.4). However, trajectory visualization for long time periods and/or large number of points leads to overcrowded displays (see Figure 4.19 (right)). This can be overcome by (i)

⁹In relation to other points in the group or other groups, e.g., distance to the closest neighbor.

showing trajectories only for a small number of recent time steps, (ii) showing trajectories only for interesting points or (iii) aggregation of movements for the whole point clouds. The first option already increases the readability of the display by reducing the previewed time period. However, it stays problematic for large point clouds. The second option needs identification of interesting points. For this reason, an *examination of movement features for points* is needed, as proposed in the Section 4.5.2. The third option is discussed in the next subsection.

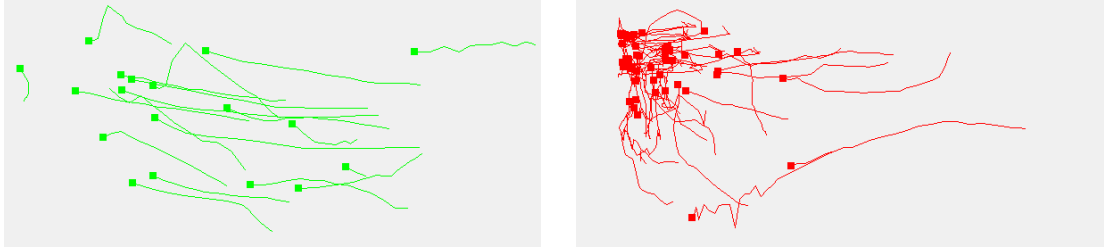


Figure 4.19.: Visualization of time-dependent points using trajectories. Left: Trajectories clearly reflect the point movements. Right: Trajectories for a complex data set leading to overcrowded display that is difficult to interpret.

4.5.3.2. Groups

Groups of entities in two-dimensions in a *static* case are represented by hulls [SSZW08,SP07]. There are various ways of abstracting a group of entities (see Figure 4.20) such as convex hull, alpha hull [EKS83,AEF*95], bounding box, butterfly hull [SSZW08], bounding circle or, newly introduced, PCA-aligned bounding box. These hulls reveal the shape of the point group however with increasing number of groups the display may be crowded. Therefore compact shapes closely following the point locations are preferred. For example, the butterfly plot largely reduces the hull area and reveals outliers. On the other hand, the PCA-aligned bounding box shows the orientation of the point cloud, the most popular convex hull shows the general shape of the point cloud including possible point combinations. These views show the shape of the data however usually disregard the point distribution. Therefore we also use the visualization of point density in combination with hulls for highlighting areas of high point concentration (see Figure 4.21 left). Additional abstractions of a group of entities used are the visualization of the center of mass of points in a group and visualization of the minimum spanning tree of the points (see Figure 4.21 center and right). The minimum spanning tree reflects the distribution of the points and their distances thereby allows for indication of dense areas and outliers (points close to the boundary with long distances to their neighbors).

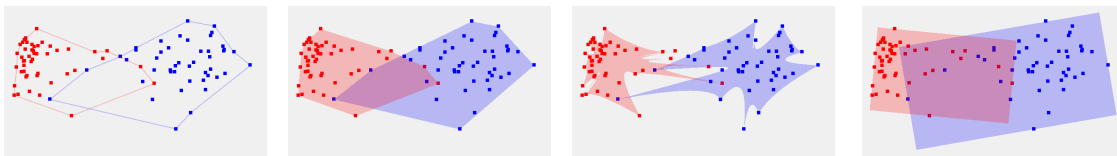


Figure 4.20.: Illustration of a variety of hulls used for point cloud abstraction. From left to right: alpha hull, convex hull, butterfly hull and PCA-aligned minimum bounding box. These hulls approximate the shape of the point cloud however disregard the point distribution within the group.

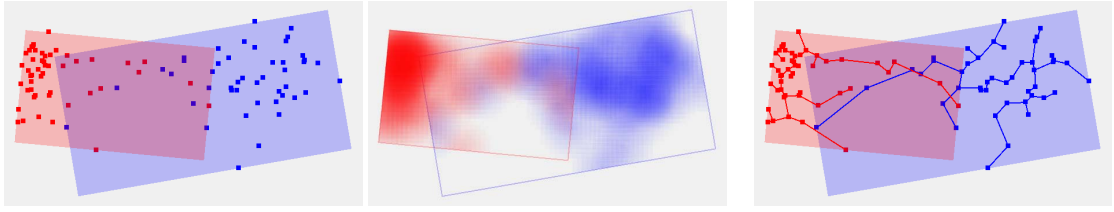


Figure 4.21.: Visualization of point cloud density. Left: Point cloud abstraction using PCA-aligned bounding box (see also Figure 4.20). Center: Density visualization in the PCA oriented bounding box using distance fields. Right: Minimum-spanning tree visualization in the PCA-aligned bounding box indicating dense areas and outliers (points close to boundary with large distances to neighbors).

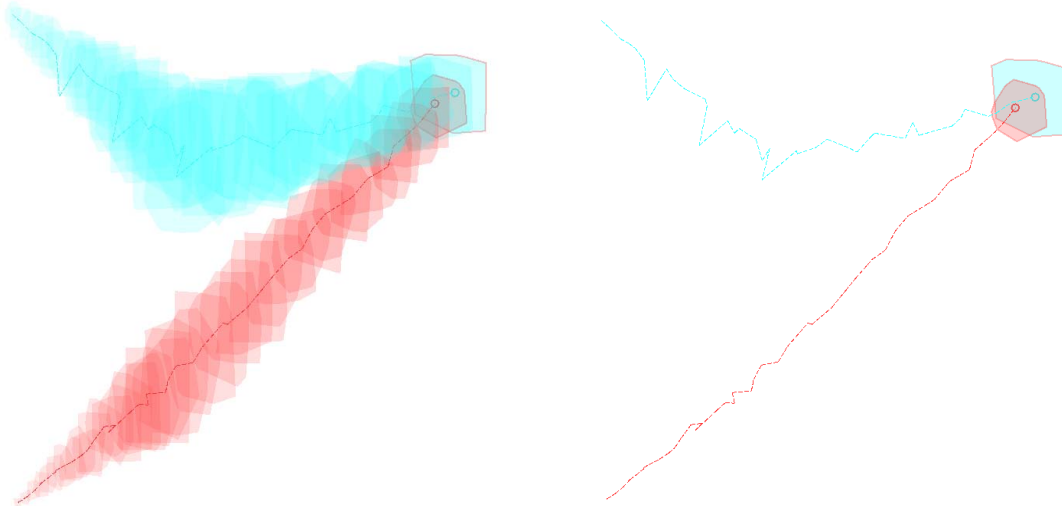
In the *dynamic* case, the changes in point cloud positions can be visualized by their path traces using transparency (see Figure 4.22 (top left)). Similarly to point trajectories, the traces can be shown for the whole time period or only for a selected number of most recent observations. Hull traces show the general movement of the whole cloud, however they disregard movements of individual points within them. Another possible abstraction of the point cloud movement is visualization of trajectories of the cloud mid-point (see Figure 4.22 (top right)). It shows the main direction of the movement of the majority of the points, however does not show differences in hull sizes and point distributions within the clouds. In an analogy to the visualization of dynamics in individual entities, however these views may, depending on the data, lead to overplotted displays that are difficult to interpret even when using mid point abstraction (see Figure 4.22 bottom left and right). Therefore monitoring of the data is used in connection to the visualization of the data.

4.5.4. Visual Analysis of Two-Dimensional Time-Dependent Data with Grouping of Entities

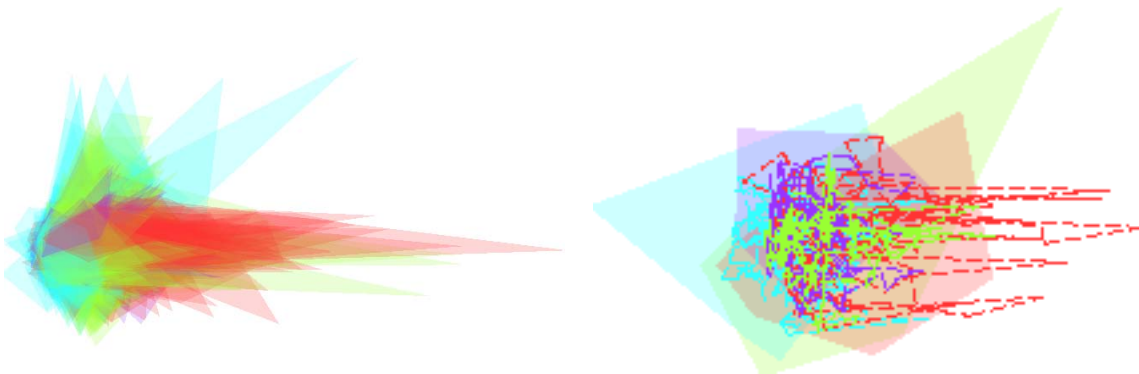
The visual analysis of the time-dependent groups of two-dimensional entities combines calculation and display of features and visualization of the input data introduced above. The calculated time-dependent features are displayed on demand as time series in line charts in addition to the main view (see Figure 4.23 for an illustration). There is a link between the views concerning the currently viewed time in the main view (indicated as a vertical line in the additional view). For monitoring multiple features and/or multiple groups, single views are provided (see Figure 4.24). The analysis of the development of certain features indicates trends in the input data. These trends can be explored in detail. These are especially useful for large and complex time-dependent data. The example below shows an illustration of the functionality of the visual analysis tool using an constructed example data set. The development of the group size, direction and overlap can be easily spotted in the monitoring windows (see Figure 4.23 right).

As the feature values may be very volatile, smoothing of features by parameterizable moving average¹⁰ can be shown instead of the original feature time series. This reveals broad trends however disregards short fluctuations (e.g., outliers). For better analysis of the data, also correlation of indicators for individual subjects (points, groups or combinations of groups) or indicators are provided. It shows which features are more distinguishable for the given data set. The combination of these tools should allow for visual analysis of complex data sets as demonstrated in Section 4.7.4.

¹⁰The moving window length is used as a parameter.



(a) Dynamics of two group of points clearly showing the movements. Left: Convex hull trace. Right: Mid-point trajectories.



(b) Dynamics of multiple groups of points showing strong overplotting. Left: Convex hull traces. Right: Mid-point trajectories.

Figure 4.22.: Two examples of visualization of group dynamics. Up: Views clearly revealing group dynamics. Down: Views leading to strong overplotting difficult to examine. Left: Traces of group hulls using convex hulls revealing the movement of the whole group disregarding the point distribution and possibly leading to overplotting. Right: Trajectories of group mid points showing the main dynamics of the groups in a clearer view disregarding the shape, size and distribution of the points.

4. Visual Analysis of Two-Dimensional Time-Dependent Data

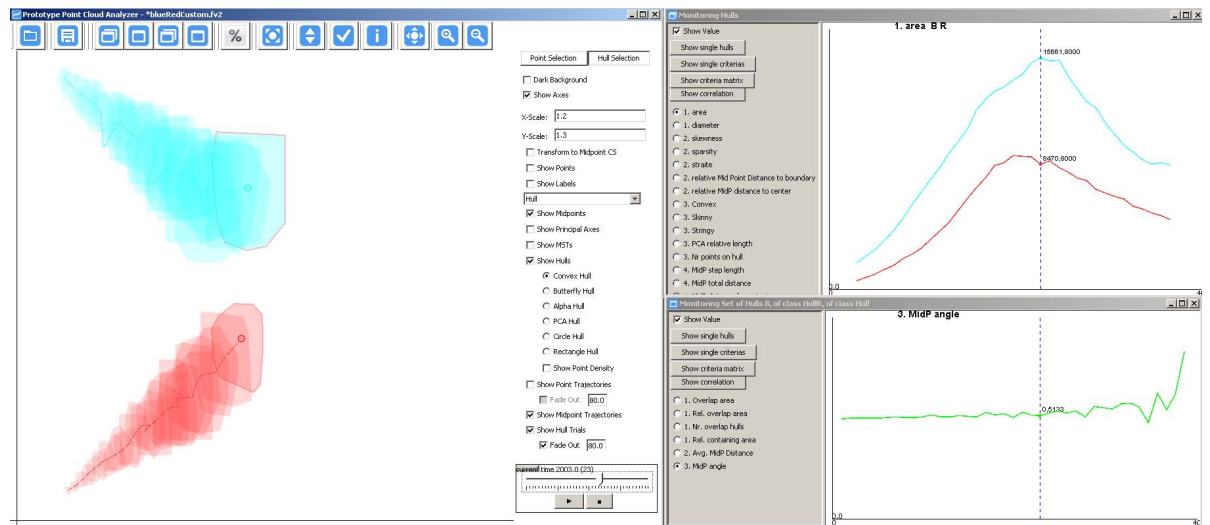


Figure 4.23.: Visual analysis of groups of two-dimensional time-dependent entities combining data visualization and analysis using monitoring of the extracted time-dependent data features. The monitoring allows for identification of interesting data patterns. Left: Window for the visualization of data dynamics with variable display possibilities. This example shows convex hull traces of two groups. Right: Windows for monitoring of the extracted features with choice of features to be monitored. The monitoring window shows an increase in groups size followed by its decrease for both groups.

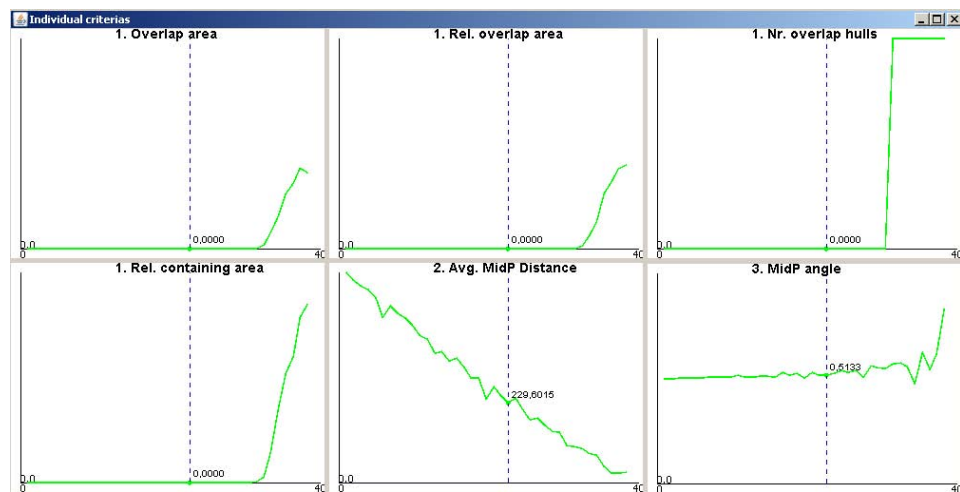


Figure 4.24.: Visualization of the monitoring of individual features using time line plots. An overview of all features over the whole time period allows for examination of the time dynamics of the individual features and interplay of the features at the same time. Interesting feature patterns and combinations of features can be discovered.

4.6. Visual Analysis of Two-Dimensional Time-Dependent Data Using SOM Clustering

4.6.1. Introduction

In this section, we present a novel approach for visual analysis of two-dimensional time dependent data (here also referred to as trajectories) based on clustering of trajectory fragments. We use the self-organizing map (SOM) algorithm for extracting patterns in the data. We offer interactive feature extraction, SOM clustering initialization and parametrization, supervision of the learning process and interactive exploration of the results and their quality. We use two types of features for determining similarity of the trajectories: direct geometric representation and abstract features.

The remainder of this section is structured as follows. Subsection 4.6.2 reviews concepts for measuring similarity between trajectories and describes selected measures used in our work. The following subsection describes interactive feature selection and interactive visual exploration of the feature space. Subsection 4.6.4 describes the interactive initialization and supervision of the clustering process. The views introduced for interactive visualization of results and their quality are presented in Subsections 4.6.5 and 4.6.6.

4.6.2. Similarity Measures and Transformation for Two-Dimensional Time-Dependent Data

We introduce relevant measures for defining similarity between trajectories. When dealing with trajectory similarity, there are several issues that need to be taken into consideration. These specifics influence the choice of similarity measure that can be applied in a specific use case.

1. *Data specifics*: The data is usually measured on discrete points in time. As there is no information on the data values between these two measurement points, therefore the data is usually interpolated between two subsequent time points (e.g., linearly).
The trajectories can be measured on regular or irregular time intervals and can contain equal or variable number of time points. In our work, we assume input data measured in regular time intervals with equal number of time steps.
2. *Tasks*: Depending on the task, we can measure spatio-temporal similarity (considering both time and space) or only spatial similarity (only considering spatial properties). Spatio-temporal similarity is a special case of spatial similarity when filtering also for co-temporal movements.
3. *Similarity measure applied*: There is a wide variety of measures that can be used. These measures are explained below (see Subsection 4.6.2.1).
4. *Data transformation to be performed*: depending on the task and use case, several transformations (see Subsection 4.6.2.2 for more information) can be performed before measuring similarity. The transformation can have an influence on the similarity result.

4.6.2.1. Similarity Measures

There are various approaches for measuring similarity between two-dimensional time series. These methods stem mainly from similarity search in trajectory databases used in geographic context. The main approaches include:

1. Using direct *trajectory geometry* in combination with a specific distance measure (see next paragraph for more details).
2. *Mapping trajectory into a feature vector* and using a feature distance measure for assessing similarity between trajectories (see below for more details).
3. *Double-cross matrix-based similarity* [KM08] – each trajectory (i.e., polyline) is represented in a double-cross matrix. Double cross matrix represents relative position of a line segment w.r.t. its starting point. It is mainly used in grid-based applications. This method disregards the absolute spatial positions of the trajectory segments.
4. *Edit distance on real sequence (EDR)* was introduced by Chen et al. [COO05]. It is based on string edit distance. It can handle trajectories of varying number of steps. It is robust to data imperfections owing to quantization of the distance to 0 and 1. However, it needs a tolerance threshold ϵ to be defined in advance. The choice of the parameter ϵ influences the resulting similarity value.
5. *Dynamic time warping (DTW)* (e.g., [Keo02]) allows for unequal number of time steps and possible phase shifts.
6. *Edit distance with real penalty (ERP)* introduced by Chen [CN04] combines L_1 norm and edit distance. It can support local time shifting and does not need the pre-setting of the parameter ϵ .
7. Distance based on *longest common subsequences (LCSS)* was proposed by Vlachos et al. [VGK02]. It gives more weight to the similar portions of the trajectory parts. It allows for stretching in time and global translation. Similarly to EDR, it however needs an ϵ threshold to be defined in advance.

In our work, we focus on the first two approaches. We explain them in more detail below. Please note that our approach can support also other similarity measures.

Geometry-based Distance Measures include several methods based on distance of points along trajectories in two-dimensional space. The simple *Euclidean distance* of trajectory points in each step in R^2 was applied by [NP06]. This measure requires equal number of points (time steps) in trajectories and is suitable mainly for equally spaced time. Pelekis et al. [PKM*07] introduce new measures for “time-relaxed” similarity, applicable also for trajectories with non equally spaced time steps. He proposes the so called *locality in between polylines (LIP)* measure defined as area between two trajectories and its variations. The variations include spatio-temporal LIP distance, directional distance, temporal directional distance, speed pattern spatio-temporal LIP distance. These measures take into account factors such as locality, temporality and directionality.

Trajectory Features Trajectory features characterize trajectories by a small number of abstract properties. The similarity between trajectories is given as distance between their feature vectors. Andrienko et al. [AAPS08] presented a set of characteristics for movement data in geographic context, which could be applied also to abstract trajectories depending on the use case. These characteristics include:

- *Length of trajectory*: measuring distance of the movement. It is of two types
 - total: $L(T) = \sum d(t_i, t_{i-1}), i = 1, \dots, n$, measures the total length of the whole movement path (i.e., sum of the lengths of all movement steps),
 - changes: measure the lengths of each trajectory segment (distance between each two following steps),
- *Duration of trajectory*: measures time duration of the movement. Please note that duration in case of equally spaced time intervals is constant in each step and is linearly proportional to the number of steps. Therefore, it can be disregarded in this case.

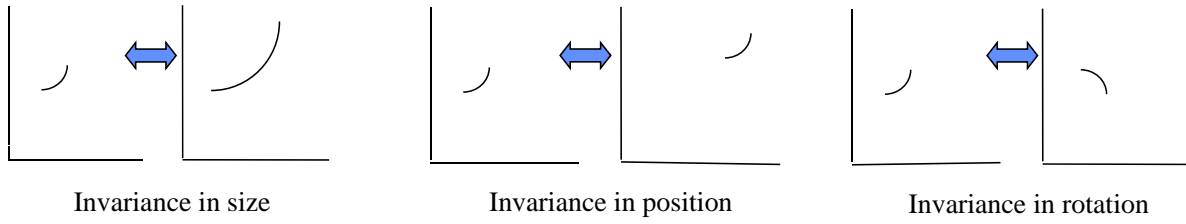


Figure 4.25.: Illustration of the three types of data invariant transformations.

- *Speed*: describes how fast an element moves ($\text{speed} = \text{length} / \text{duration}$). Please note that in case of equidistant time steps, this measure is linear transformation of the length measure, therefore can be disregarded,
 - total speed: is analogy to length for equally spaced time intervals,
 - speed changes: is analogy to length for equally spaced time intervals,
- *Direction*: reflects the direction of motion,
 - major direction: reflects the general direction of the movement measured as the direction between start and end point of the movement,
 - Dynamics of direction: is an analogy to above, measures directions of each two subsequent steps. It can indicate major turns, or straight movements.

These features are applicable to any two-dimensional time dependent data although they have been developed for geographic movements. In our work, we concentrate on the specific type of two-dimensional time series with equidistant time steps and equal number of time steps. Therefore some of the features are dispensable. In particular, the duration is constant and speed characteristics can be represented by length characteristics when time intervals are equidistant.

4.6.2.2. Transformation

When dealing with abstract two-dimensional time dependent data, scaling, rotations and translations of the two-dimensional time-dependent data can be applied as a pre-processing step. The type of transformation depends on similarity notion in a particular application and task context. The transformation used thereby impacts the results of similarity measures presented above. Figure 4.25 presents three types of invariance that could be of interest. Please note that these transformations usually are not applied for geography-based movement data as the fixed point locations play a significant role in the notion of similarity.

Figure 4.25 shows three possible invariances that can be used for data transformations: Please note that these invariances can be combined in particular use cases.

- *Invariance in size*: is used when the trajectory length is not important, for example, when it is not important how big the movements are and the focus is on the shape of the movement.
- *Invariance in position*: This applies when the exact location is not important and translations in space are possible. In this case, only relative movement is of relevance. For example, it is not important where exactly the movement takes place, the movements are then compared relative to each other.
- *Invariance in rotation*: In this case, the total direction of movement is not relevant. This is used, for example in handwriting recognition.

Based on the above-mentioned invariances, transformations of the trajectories (i.e., translation, normalization and rotation) are undertaken. These transformations can be performed on the global, local, entity, or time level. For example in size normalization, global level means normalizing over all trajectories of all entities over the whole time period, local means normalizing each trajectory separately, entity-based means normalizing all trajectories of each entity together and time-based is performed for all entities in each time period. For example, for risk-return data, we can normalize each asset separately or over all assets on the market either during the whole time period or for each week separately.

4.6.3. Interactive Feature Selection and Visualization of Feature Space

In our work, we consider both direct geometric representation of trajectories and abstraction of trajectories into feature vectors. We apply Euclidean distance, as our dataset is equidistantly spaced with the same number of time steps. If needed, the trajectories can be first partitioned into equidistantly spaced fractions with the same number of time steps. For example, the risk return data can be divided into full trading weeks. We note that this type of distance has been also used for clustering of trajectories by Nanni and Pedresci [Nan02] and [NP06]. Thus the usage of the same similarity function increases comparability of their results with ours.

1. Geometric representation: We use a simple trajectory vector representation constructed from normalized trajectory sample points. To obtain the vector representation, for the purpose of the target financial application, we first normalize each trajectory by scaling it into the unit square $[0, 1]^2$, and then sample n uniformly spaced (x, y) coordinates spanning the trajectory from its start point to its end point. The concatenation of the sample coordinates in their sequence along the trajectory yields the vector representation of length $2n$. $T = \{x_0, y_0, \dots, x_n, y_n\}$. By definition this representation ignores features, which might be important in certain applications. For instance, it ignores the trajectories' absolute positions and scale in space, and, depending on the number of samples, may lose trajectory details or introduce sampling artifacts.

The key advantage of the geometric data representation in the context of this work is that it has a direct geometric interpretation and therefore can be directly displayed in a 2D plane. It is suitable as the basis for direct visualization of and interaction with cluster prototype vectors produced by the SOM algorithm. Besides, this vector representation is simple to obtain. As the $2n$ vector of geometric coordinates can be interpreted as a feature vector, it allows for a straightforward calculation of trajectory distances.

2. Abstract features Based on the normalized trajectories (geometry-based feature vectors) presented in the previous paragraph, we calculate the following additional features motivated in Subsection 4.6.2.1: minimum, maximum, average and total both for step length and direction. Additionally the direct length between starting and end point are calculated. These features can be interactively selected and weighted using visual interface (see Figure 4.26). The user interface and its functions resemble those of graph feature selection presented in Section 3.6.4. The feature selection (weighting) is supported by automatic identification of highly correlated and irrelevant features. The correlation between features which can be also visually inspected in the correlation matrix. Note that the sum of feature weights is normalized to 1.0 and automatically recalculated by each weight change. Moreover, the calculated feature space can be examined using feature distribution view (see Figure 4.27). Features are deemed as irrelevant, when their variance equals zero (i.e., they are constant).

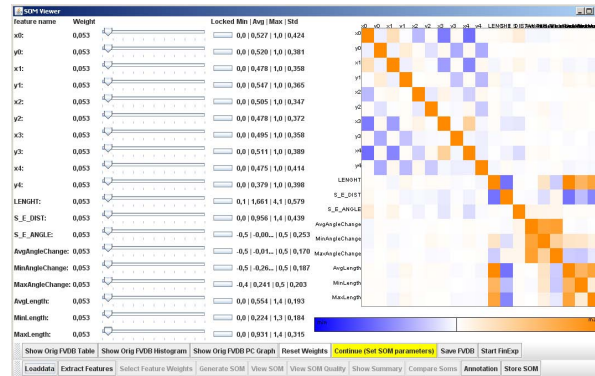


Figure 4.26.: Interactive selection and weighting of trajectory features determining the similarity calculation. From left to right in the picture: Feature names, current feature weights, weight adjustment interface, basic feature statistics, feature correlation matrix with color scheme from blue (-1) to orange (1).

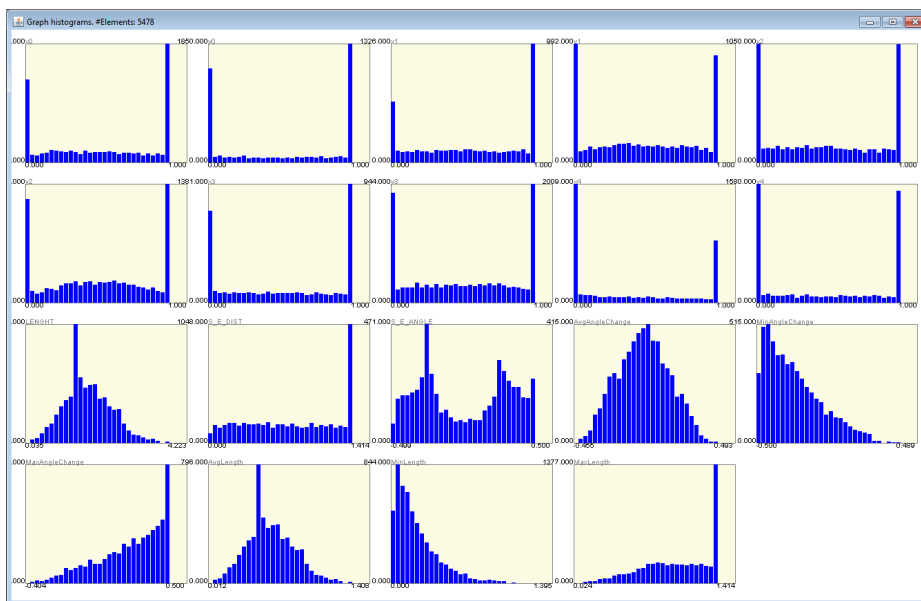


Figure 4.27.: Visual exploration of feature space using feature distributions. The histogram of each feature is displayed in order to assess the distributions of the feature values across the whole data set and indicate suitable feature selections.

4.6.4. Interactive Visualization and Control of SOM Clustering Process

We now present advanced features for SOM training. Before the SOM training process can start, the grid of cluster prototypes needs to be initialized. The initialization guides the training process, and often influences the overall layout of the emerging cluster map. In the standard approach, two initialization methods are common: Random initialization, and initialization based on a Principal Component Analysis of the input data set [Koh01]. Both methods are unsupervised in nature.

In this section, we describe the implementation of a user-driven interactive initialization of the SOM grid, which can be used for advanced users or when partial knowledge of the data structure is available. After setting-up the SOM training parameters, the learning process is usually performed without user supervision. However, in our approach, we provide tools for interactive visualization of the training process and its control. The controlling feature is aimed for advanced users.

4.6.4.1. Map Initialization Based on Trajectory Editor

As an advanced feature, we use a more user-oriented control of the initialization process. We base the approach on the fact that trajectory data have a straightforward geometric representation and can therefore be readily visualized and manipulated interactively. To do so, we provide an interactive *trajectory editor* that lets the user draw example trajectories into chosen SOM grid positions. Reference trajectories may be input at distinct map locations, thereby specifying a model for the overall SOM cluster layout desired. Starting from a user-provided set of example trajectories, we initialize the full grid of SOM trajectory prototypes as follows:

- for the grid nodes for which the user has provided example trajectories, we set the initial value of the SOM prototype vectors,
- for the unassigned grid nodes, we interpolate between the assigned example vectors.

Please note that when using geometry-based similarity measures, the feature vectors directly encode the trajectory geometry (the sequence of trajectory control points). When using abstract trajectory features for determining similarity, the initial geometric representation created by the user needs to be converted into a feature vector.

Figure 4.28 shows an example of the trajectory editor for initialization of the SOM prototype vectors. Five reference trajectories were assigned by the user, and the remaining prototype vectors were filled in by weighted average interpolation. With this concept, the user is able to efficiently initialize a SOM prototype map with a coarse template of a desired layout. For the interpolation a chosen algorithm from a set of possible algorithms, such as nearest neighbor or weighted average, can be used.

4.6.4.2. Online Visualization and Control of the Map Training

Visualization of the Training Process Recall that in our application, the data vectors have an immediate geometric interpretation. Therefore we are able to visualize the online training process by showing a continuously updated display of prototype trajectories. Specifically, the user can observe the effect of the provided trajectory initialization on the subsequent training process. In addition to visualizing the emerging trajectory patterns within the SOM cells, we optionally superimpose certain cluster map quality metrics using color-coding and nearest neighbor connectors (cf. Figure 4.29):

1. Color-coding of the current quantization error of the emerging maps: For each prototype vector, we calculate the average Euclidean distance between the prototype, and the trajectory data samples it represents. It allows for assessment of the actual values of clustering quality and their development during the training process revealing whether the process evolves well (decreasing the quantization error).

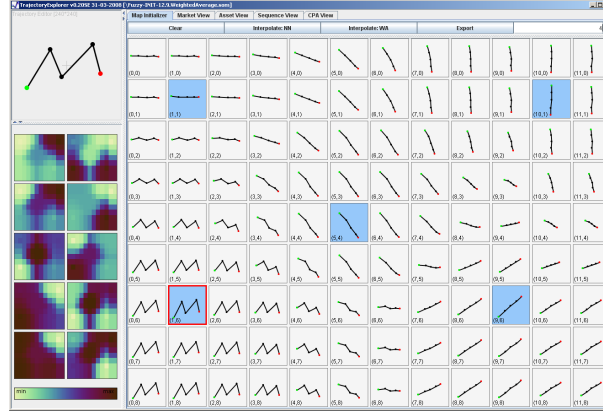


Figure 4.28.: Editor-based initialization of a 12×9 SOM trajectory grid, using 5 user-defined example trajectories (marked blue) in conjunction with weighted average interpolation. Component distributions (x_0, y_0) to (x_4, y_4) are shown in the left panel.

2. Color-coding of the average Euclidean distance between each SOM prototype vector and its immediate prototype vector neighbors on the grid (also known as U-Matrix [Ult03] color coding) [Ves99]). The U-Matrix representation indicates cluster areas and “borders” between neighboring areas of the SOM grid.
3. Nearest-neighbor connectors indicating the nearest neighbor relations between the SOM prototype vectors. This visualization reflects the smoothness of the pattern transitions over the map (smoother transiting prototype layouts show shorter connectors). This indicates the clustering quality according to topology preserving criteria.

By means of these visualizations, the user can observe both the emerging organization of the pattern layout, as well as the quality of the representation of the obtained clustering. Figure 4.29 illustrates the on-line training visualization with snapshots of the quantization error during training of a 12×9 SOM of trajectories (a-c) and a zoom into a connector display (d).

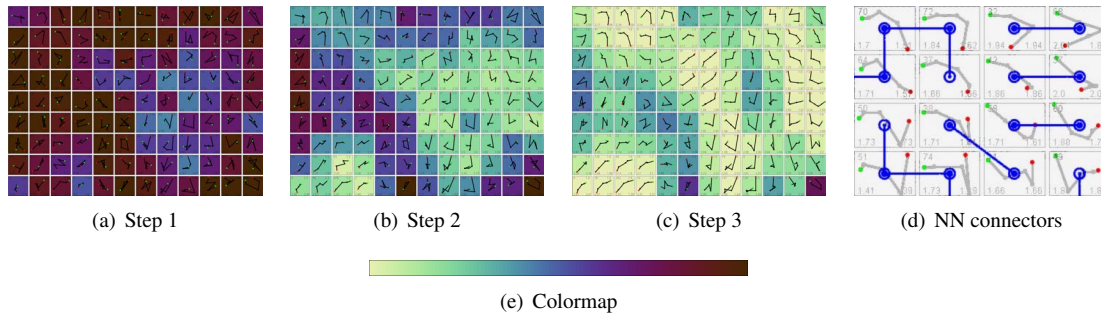


Figure 4.29.: Visualization of the online learning process by color-coding of the quantization error (a-c; brighter is better). Nearest neighbor connectors (d) are an optional overlay indicating the smoothness of the trajectory pattern transitions over the trajectory map. The connectors show the nearest neighbor relationships between the reference trajectories (shorter is better).

Control of the Training Process In order to be able to control the training process, a set of interaction facilities is needed. In this way, at any time, the training process can be suspended and, depending on preferences and user experience one or more of the following controls can be executed:

1. Adjust single prototype trajectories by directly editing them with the trajectory editor.
2. Adjust the map by editing a selection of prototypes and replace the remaining prototypes by interpolating between the selected prototypes.
3. Update the training parameters: Adjust the number of remaining iterations, learning rate, and neighborhood kernel for the whole grid or for selected cells.
4. Reinforce training of selected patterns.

These controls serve to guide the learning process toward user desired results, if required. These advanced options are designed for fine-grained control of the training by experienced analysts. After updates to the training process have been manually entered, training is resumed and the observation of the effects is possible. Usually, experimentation with different parameter settings is required for optimizing results on a given data set and analysis task.

Controls 1 to 3 have been explained in the previous text. Control 4 provides the specification of smaller or even zero learning rates for selected cells. This allows to explicitly enforce patterns in selected cells on the map. Note that the idea of fixing selected data vectors to given SOM grid locations during training is not new per se. For instance, the SOMPAK implementation includes an option for doing so [KHKL96]. We point out that our interactive training controls extend beyond a simple fixing of vector assignments. Not only basically any training parameter may be edited at runtime, but also, the reference vectors may be interactively modified during training using the trajectory editor.

We also point out that, in principle the control framework allows a user to produce any prototype layout desired, possibly influencing the reliability of the obtained results. Generally, we expect that an application- or user-dependent trade-off will have to be found between supervised and unsupervised training of the reference map. Clustering quality visualization is recommended for appropriately balancing the trade-off between the precision of the clustering (in terms of quantization error and nearest neighbor transition) on the one hand, and supervised pre-assignment of the reference layout on the other hand.

4.6.5. Interactive Visualization of SOM Clustering Results

The interactive visualization of clustering results provides tools for visual exploration of the data set. As a basis for the various data views, the visualization of trajectory samples and cell members in SOM grid is used. The SOM grid provides a representation of many trajectory fragments with a smaller number of trajectory prototypes. The size of the SOM grid is an algorithm parameter. In each grid cell, these prototypes can be displayed. For the graphic representation of the patterns, we use their direct geometric representation when using geometric features for clustering and abstract feature vector representation as parallel coordinates or display of the trajectory nearest to the cell center in the other case (see Figure 4.31 for an illustration).

Based on the above mentioned SOM grid displays, four views are built (see following subsections for more details).

- The *general view* allows to assess overall distribution of the patterns identified in the data set.
- Restricting the general view to a selected object results in a so called *object-oriented view*, which allows the analysis of the distribution of movement patterns occurring for a given object over the whole time period.

- The *time-oriented view* is an analogy to the object-oriented view, where the restriction has been posed on the time span. It shows the distribution of patterns for a selected time span over all objects in the data set.
- *Pattern sequence view* is a comparative view on the time-dependent sequence of patterns of many objects simultaneously. It allows specifically to search for co-occurring and correlated patterns among time and objects.

The general view can be used for a wide variety of data sets. Time-oriented and sequence view are suitable for time-dependent data with multiple time spans and object oriented view is favorable for data sets with multiple objects.

In addition to these views, also the presentation of the so-called *component planes* (see Figure 3.37) is provided. It shows the values of the trajectory features across the SOM grid. The values are displayed as a heatmap.

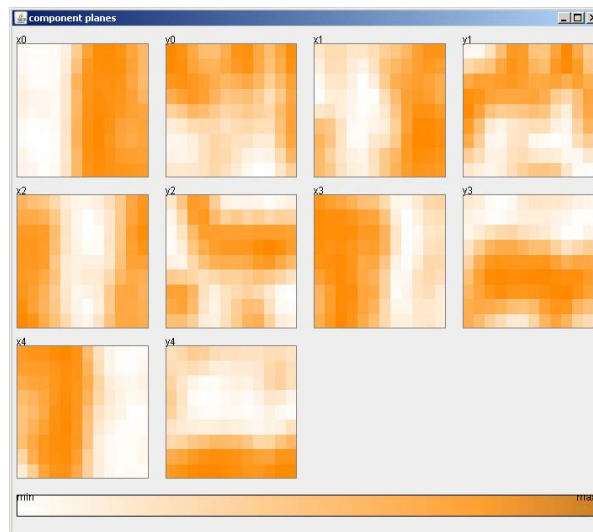


Figure 4.30.: Visualization of component planes for analysis of feature distribution across the SOM grid. It shows the values of each feature in the cell center across the SOM grid using a heatmap matrix. White color indicates low values and orange color means high values.

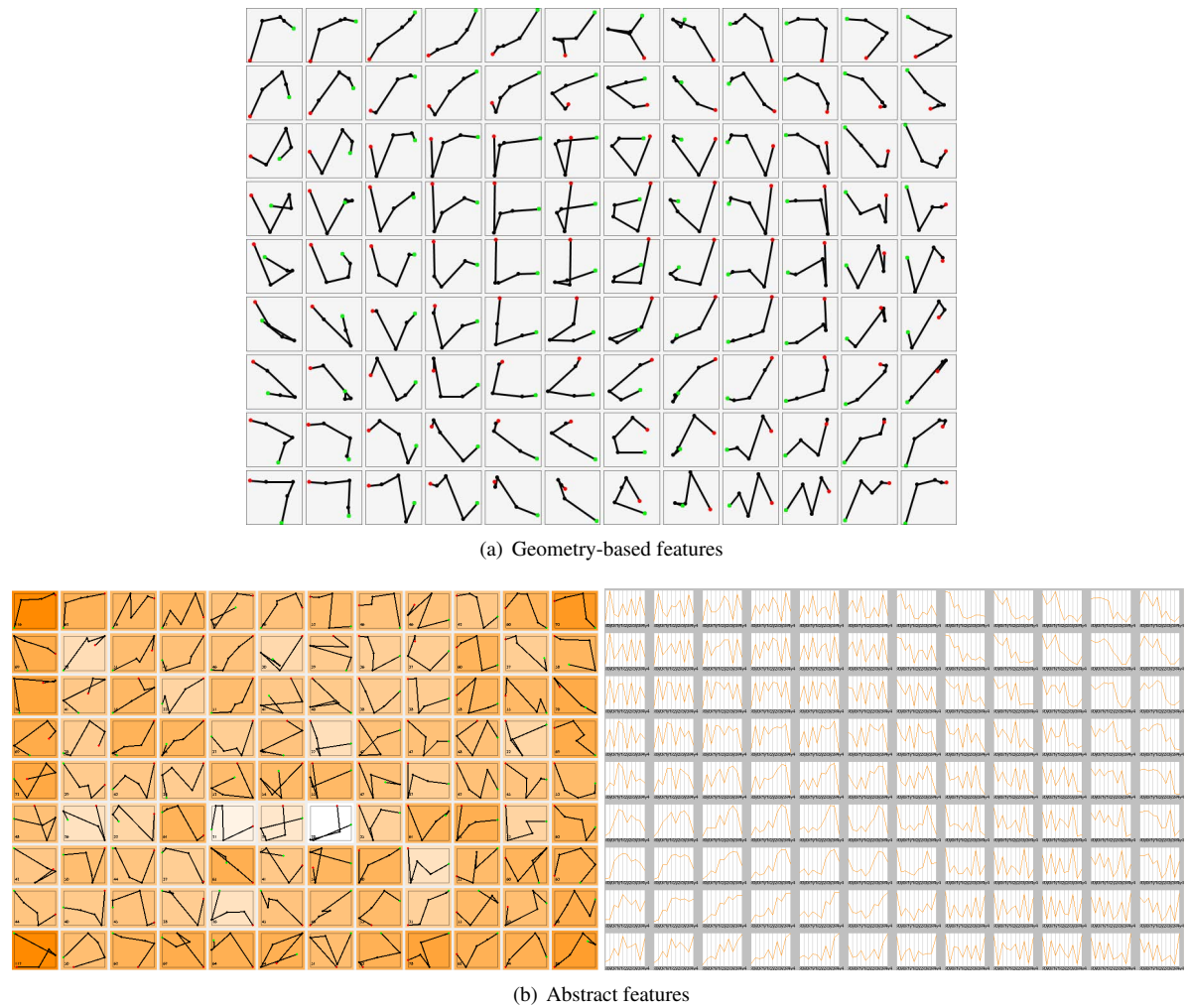


Figure 4.31.: The SOM result view in cell grid. Top: Trajectories of the cell prototypes when using features with direct geometric representation. Bottom: Two types of result view when using abstract features. Bottom left: showing a trajectory nearest to the cell center. Bottom right: Abstract parallel coordinate view of the cell center.

4.6.5.1. General view

The general view displays the distribution of movement patterns of all objects and over the full time horizon (see Figure 4.32). In conjunction with the SOM clustering process, it yields an effective overview of the general-characteristic patterns in the data set. When using color coding for the background depending on the number of matches for a cell (cell members) (in the same way as in SOM result visualization), also the frequency of patterns can be explored.

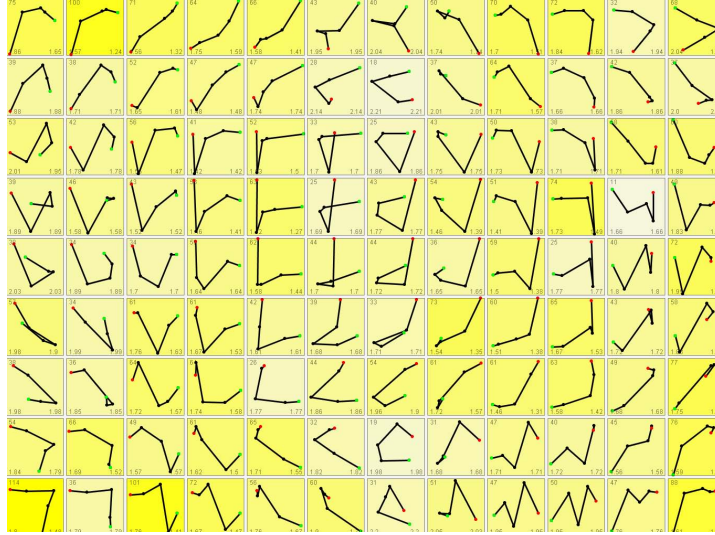


Figure 4.32.: The general view visualizes the distribution of data movement patterns in the data set. Owing to the topology-preserving properties of the SOM algorithm, the map can be meaningfully interpreted in terms of pattern transitions. The background color (from white to yellow) indicates the frequency of the found patterns.

4.6.5.2. Object Oriented View

The object-oriented view (see Figure 4.33) is obtained from the general view by restricting the set of sample trajectories to a selected object. Thereby, individual objects can be analyzed for occurrence of specific data movement patterns. Practically, due to the small number of matches per prototype, we directly overlay each matched sample to its prototype, and keep the patterns not matched by any sample in context by rendering only the prototype in a lighter color. The dashed line represents the median. For the matched prototypes, we use a coloring of the background to indicate the number of represented samples, scaling the color saturation proportional to the maximum occurring frequency.

Additional features in the object-oriented view support visual analysis of the *temporal sequence of patterns for a selected object*. As each object trajectory fragment can be matched to a prototype trajectory, the sequence of data movements (i.e., the sequence of the fragments) can be visualized by connecting the respective SOM prototype positions. By restricting the distance between two consecutive movements in time, it is possible to filter for gradual or abrupt inter-temporal pattern transitions. This is demonstrated in Figure 4.34, which shows abrupt inter-temporal pattern sequences for several objects, by filtering for a minimum grid distance of 14. The



Figure 4.33.: The object-oriented view is a version of the general view (see also Figure 4.32) restricted to a given entity of interest. It shows the distribution of patterns occurring for a specific entity, and over the full time span.

view allows the identification of time spans where the object movement in the following time span roughly reversed. Filtering for small distances on the other hand would reveal periods of roughly recurring patterns over time, or smoother transitions thereof.

We state that depending on the selectivity of the filtering, overplotting effects could arise in this basic line-based view. A solution would be to rely on more advanced approaches for visualization of larger numbers of pattern connectors. To this end, an adoption of the edge bundle technique [Hol06] seems promising.

4.6.5.3. Time-Oriented View

A simple yet powerful view is achieved by filtering the set of sample trajectories by user-defined time subintervals. Thereby, the user can easily obtain an understanding of the distribution of patterns over time. This visualization technique follows the object-oriented view (see Figure 4.35). It also uses the same background coloring scheme to show the number of samples for each pattern occurring in the selected time period.

4.6.5.4. Sequence View

The sequence view is a comparative view of movement patterns, for all entities over all time periods. The view is organized in a row-by-column scheme where each row refers to an object, each column refers to a given time fragment, and each cell contains the prototype representation of the actual movement sample. It provides an overview of the sequences of movement patterns over time and movement correlations between objects.

One main use case of this view is visual analysis for correlations, co-occurrences, and frequency of patterns. For example, it allows to detect recurring patterns in individual time points or similarly moving entities over the time period. To this end, an automatic preprocessing of the sequence view can be undertaken using statistical evaluation methods. The results of this analysis are input for the visualization, controlling filtering and highlighting. In particular, we used a two-stage analysis scheme for automatic identification of possibly interesting view

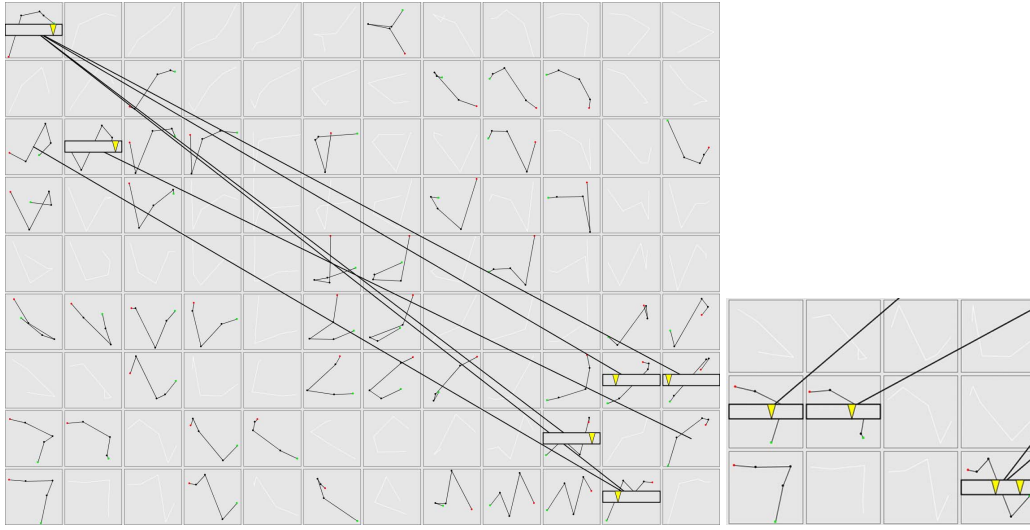


Figure 4.34.: Object-oriented views showing abrupt inter-temporal pattern transitions by connecting the SOM cells with consecutive trajectory fragments. Time tick scales are used to indicate the date of occurrence of the pattern transition, relative to the global time scale (see bottom-right for a closeup).

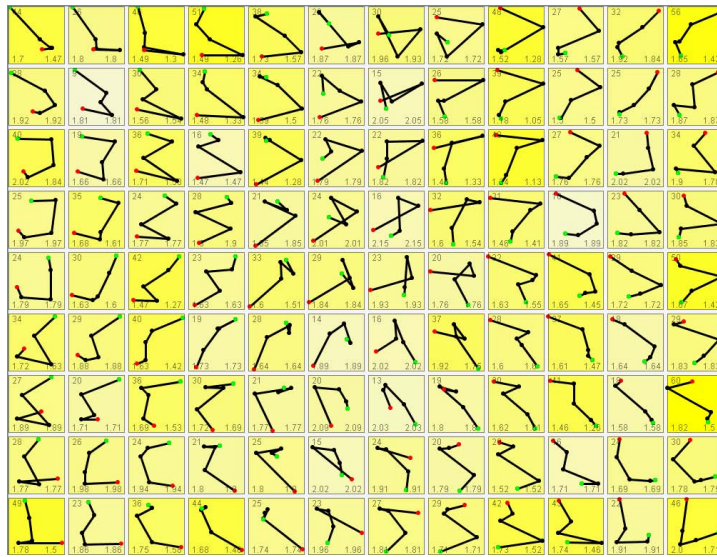


Figure 4.35.: Time-oriented view shows the filtering of the general view to a particular time fragment. It shows the distribution of patterns for a specific time period fragment.

configurations. Firstly, the algorithm calculates the entropy measure [EM95] for the distribution of movement patterns across time fragments. In the second stage, an analysis of pattern frequency considering the grid-based pattern distances finds the most prominent patterns in the time spans of lowest entropy. These patterns represent those patterns in the identified time periods with similar data dynamics.

The results of the entropy analysis are presented to the user in form of thumbnail sequence views. To easily spot time periods with dominant market dynamics, a sorting of rows (objects) according to distance from the market trend is undertaken before the results are visualized. In the views, the prominent patterns are highlighted. Our highlighting scheme assigns highlighting color saturation to reflect the similarity of each sequence pattern to the identified prominent pattern. Specifically, we assign three highlight color saturation grades centered around the selected pattern as shown in Figure 4.36 for two different patterns.

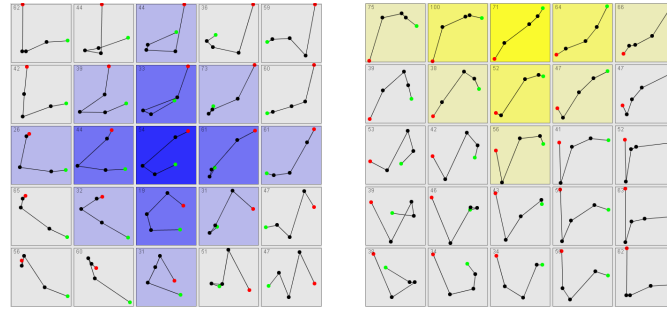


Figure 4.36.: Two trajectory patterns located at positions [6,6] (blue) and [0,2] (yellow) on the general view displayed in Figure 4.32. Neighboring patterns with grid distance 1 and 2 are highlighted at decreasing saturation. This color-coding is used in the sequence view shown in Figure 4.37.

Figure 4.37 shows the sequence view with two selected patterns highlighted. It is the most detailed view which at the same time, suppresses small entity-specific detail and noisy patterns. The full sequence view supports visual analysis of the distribution and correlation of the two selected patterns over the full time period. Owing to the large amount of input data, the detailed sequence view requires a high resolution display. Therefore, the sequence view is also an interesting application for usage with large-scale displays such as the HEyeWall[®] [HEy] or PowerWall [Pow] systems [TSB*08]. An illustration of such an application can be seen in Figure 4.38.

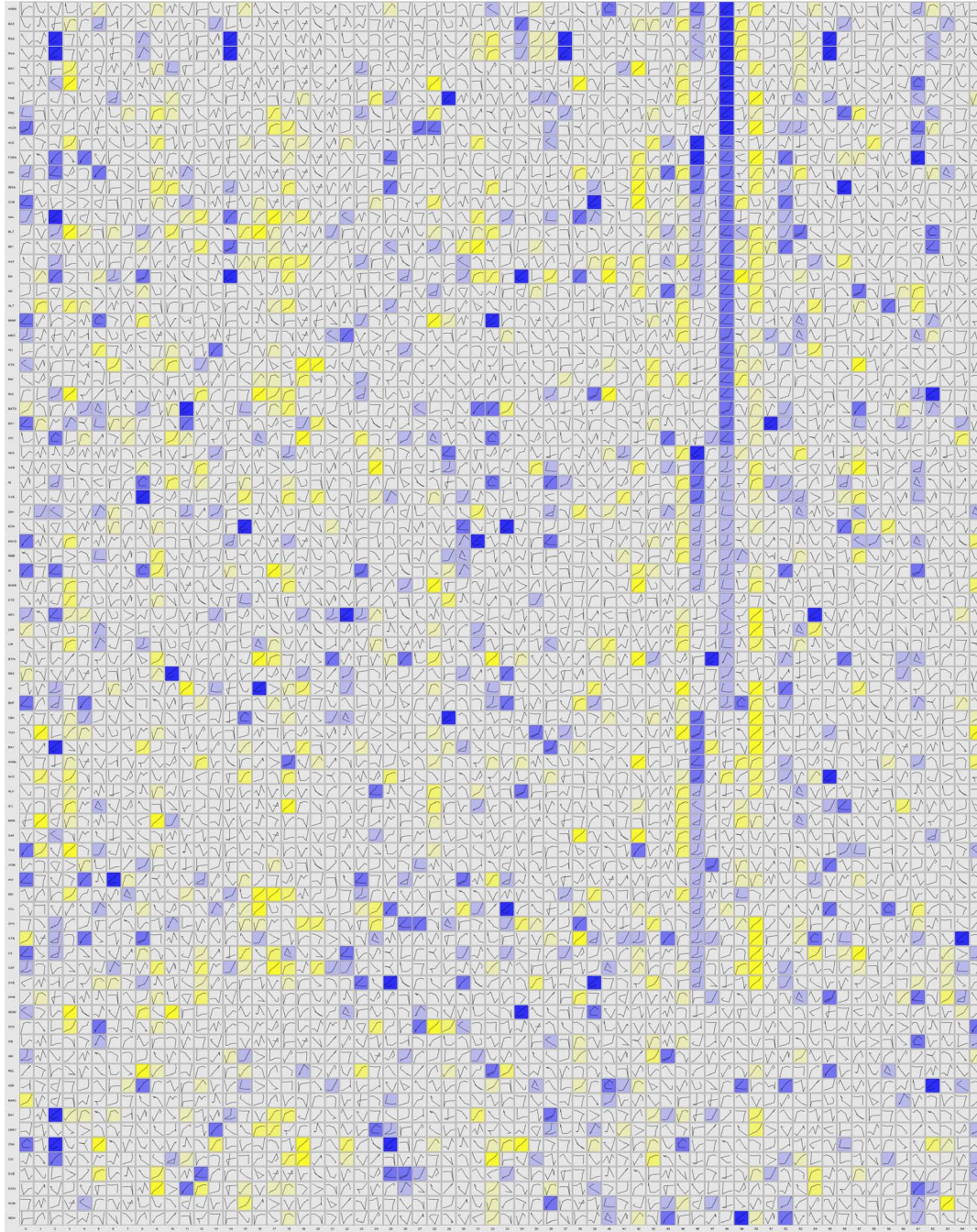


Figure 4.37.: The sequence view visualizing data movements of 83 objects during 66 time spans of observation. Patterns [6,6] and [0,2] as selected by the user from a set of automatically generated candidate patterns are highlighted (see also Figure 4.36).

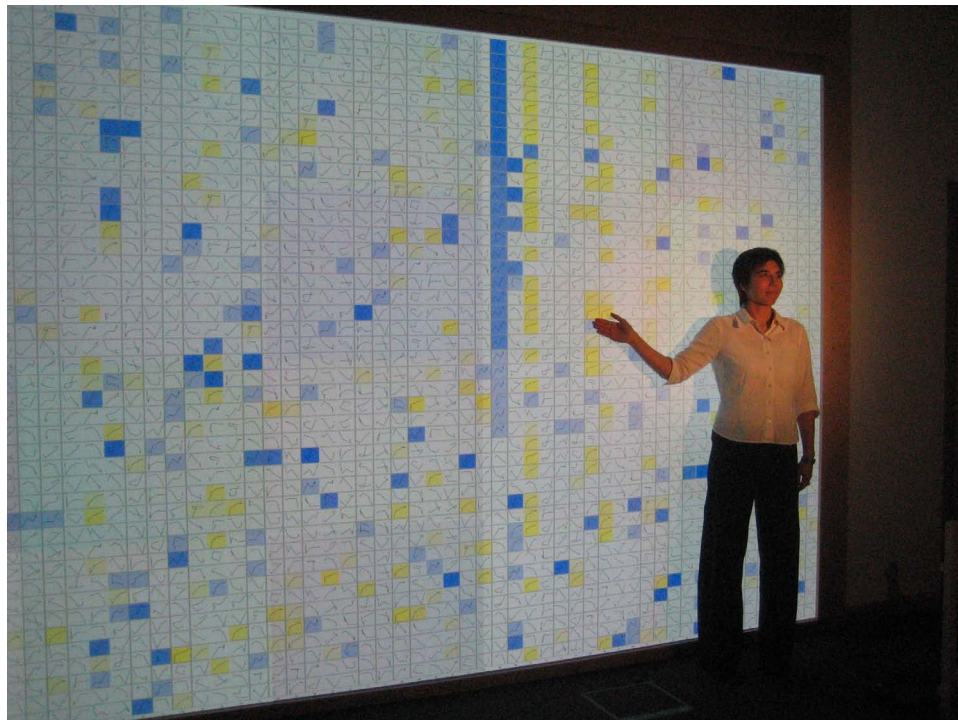


Figure 4.38.: Application of sequence view on the HEyeWall large-scale display system.

4.6.6. Interactive Visualization of SOM Clustering Quality

In order to support the assessment of the clustering quality, several interactive views based on feature and distance distributions, difference to cell centers and calculation and visualization of clustering quality measures are provided. These views are an analogy to the views presented in the section on graph clustering (see Section 3.6.6) therefore we do not discuss them in detail and concentrate on the specific views for two-dimensional time-dependent data.

4.6.6.1. Interactive Visualization of distributions of distances to cell center and cell member features

As an indicator for clustering quality, the distance of cell members to its center can be examined. The distribution of the distances across the grid can be seen in the *distance distribution view* (see Figure 4.39). In the same way as in graph clustering, outlying members can be displayed in a separate view, where the background color corresponds to the distance to the cell center. In this view, the distances should be close to zero for good clustering results (the histograms should be strongly left skewed).

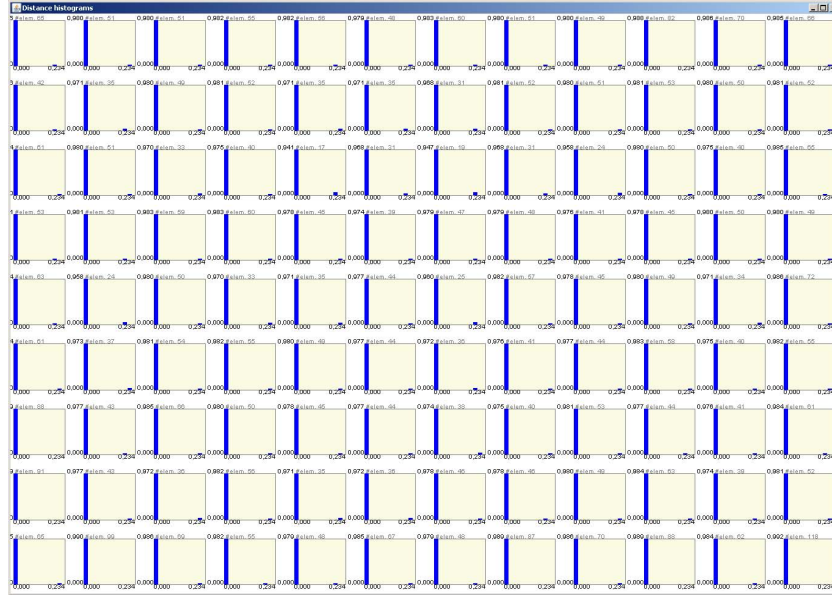


Figure 4.39.: The distribution of distances between cell members and cell center across the SOM grid are displayed in form of a histogram. Small distances indicate good clustering results therefore largely left skewed distributions close to zero are preferred (as shown in the figure). Larger distances at the end of the distribution indicate outliers. These cluster outliers together with cluster representatives can be displayed in a separate window on demand (see also Figure 3.40 on page 95).

Alternatively, the *feature distributions* for the members of a cell (see Figure 4.40) allows to analyze the compactness of the clustering and spot outliers in the cells. Similarly to the graph clustering, on demand, an overview of the cluster members across the feature distribution or parts of the histogram can be displayed.

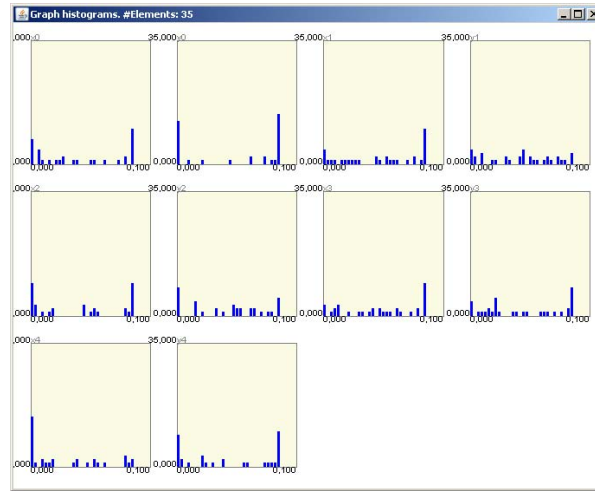


Figure 4.40.: Visual analysis of feature distributions for a particular SOM grid cell. The histograms of each feature in a cell are shown. Narrow distributions are expected for good clustering results. Wider distributions may indicate the need for inclusion of this feature in further clustering. On demand, an overview of the cell members, showing representative trajectories from all parts of the selected feature distribution, can be displayed in a pop-up window (see also Figure 3.41 on page 96). This allows for inspection of the data variance with respect to the selected feature.

4.6.6.2. Direct Visualization of Trajectory Difference to SOM prototypes

Another option for clustering quality assessment is the appropriate visualization of the distribution of trajectory samples with regard to a prototype trajectory. This view shows how well the trajectories in a cell match the sample trajectory (i.e., the cluster center). Depending on trajectory feature vector construction, we can either use direct geometric trajectory representation (so called “*trajectory bundle*”) or an abstract view (so called “*abstract bundle*”) (see Figures 4.42 and 4.43).

Trajectory Bundle Visualizing a prototype trajectory together with a few associated samples of limited length is straightforward. For example, for a given time period and small subset of objects, we may simply overlay the respective polyline paths over the associated prototype p , connecting corresponding path segments by dashed alignment lines (see Figure 4.41 (a)). We indicate start and end points of the prototype trajectory fragment using green (start) and red (end) markers.

Overlaying more than a few sample trajectory fragments, the display quickly gets crowded. We therefore resort to omitting rendering of the polylines directly, but instead use a coloring scheme such that color intensity reflects the density of trajectories around the given prototype. We fill the (possibly self-intersecting) polygons given by each pair of corresponding prototype and sample trajectory segments (see Figure 4.41 (b)) using a basic semi-transparent color. Specifically, we set the transparency proportional to the overall number of trajectory samples to visualize. The effect is that areas sharing much segment overlap get colored more intensively, naturally communicating trajectory density information by a transparent “veil” around the prototype trajectory. We like to call this technique the *trajectory bundle* visualization, noting its scalability for large numbers of trajectory samples. Figure 4.41 (c) illustrates a trajectory bundle consisting of 54 samples.

We point out that the trajectory bundle visualization has a foundation in an analytical trajectory dissimilarity metric, as it can be interpreted as a visual generalization of the recently proposed *LIP* trajectory dissimilarity metric [PKM*07] to *sets* of trajectories. The technique was also inspired by the so-called *opacity bands* visualization originally proposed for parallel coordinate plots [FWR99].

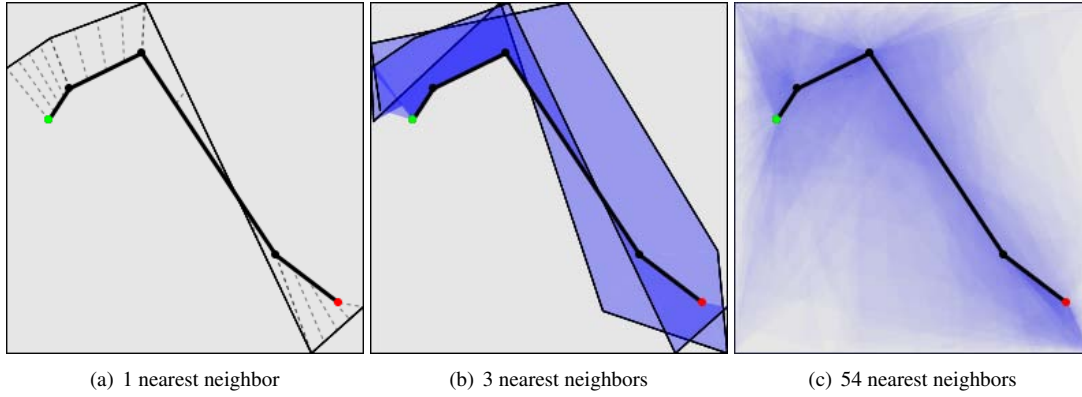


Figure 4.41.: Prototype trajectory (thick polyline) and one (a) and three (b) nearest neighbor (NN) sample trajectories. The *trajectory bundle* visualization (c) uses an overlay of transparently colored segments to indicate the spatial distribution of many trajectories simultaneously.

Abstract Bundle In an analogy to the direct trajectory representation, the bundle views can be applied also to abstract views using bundles around parallel coordinate representation (see Figure 4.43). The bundle visualization technique described above is directly applied to the parallel coordinate view.

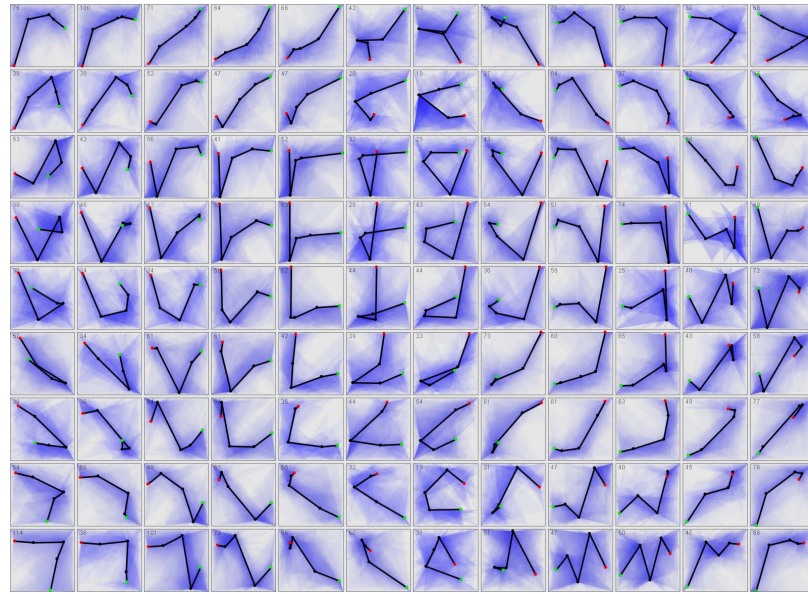


Figure 4.42.: Trajectory bundle visualization shows the difference between actual cell members to the cell center (sample trajectory) in the SOM grid when using geometric trajectory features.

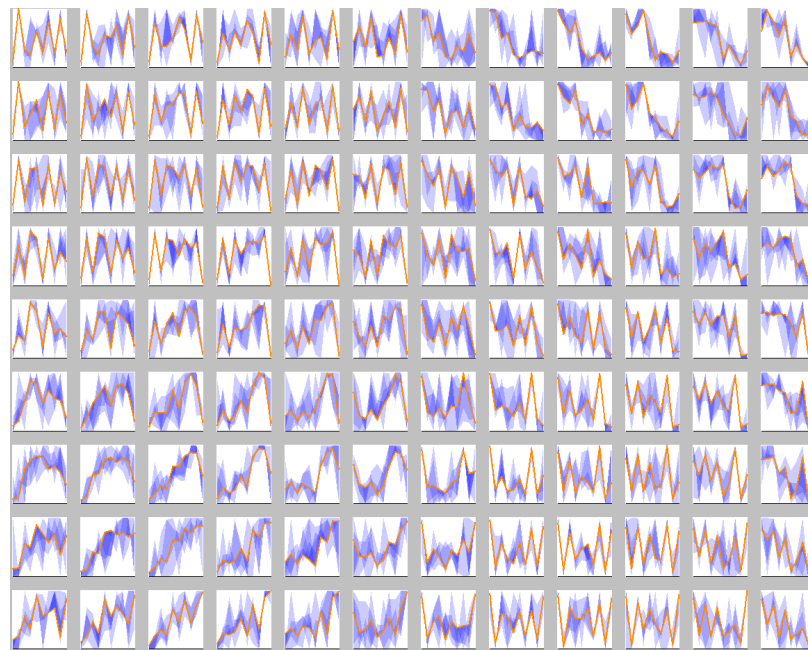


Figure 4.43.: Abstract bundle visualization shows the difference between actual cell members to the cell center (sample trajectory) when using abstract trajectory features as an analogy to the trajectory bundle visualization in Figure 4.42.

4.6.6.3. Visualization of algorithmic assessment of clustering quality

In addition, the previous views can be combined with the algorithmic *assessment of clustering quality by various measures*. The measures are based on clustering quality literature (see Subsection 2.5.2.1) also used for assessment of clustering quality in Section 3.6.6. The values of a selected measure across SOM grid can be displayed, if applicable, as colored background of the cells (see Figure 4.44 for an illustration using quantization error measure).

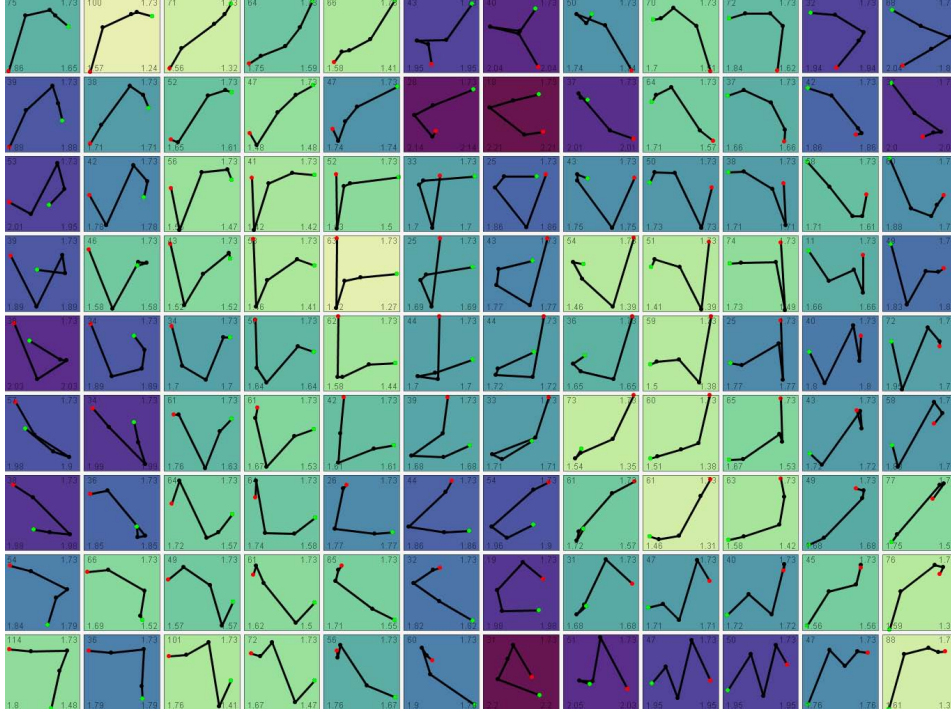


Figure 4.44.: Visualization of a selected SOM quality measure. The quantization error shows the average distance of cell members to the cell center. The measure values are displayed in the SOM grid using heatmap. Bright colors mean low distances (good) and dark colors indicate cells with high distance (worse). This view shows good overall clustering quality with one cell with larger distance that can be inspected in more detail on demand.

4.7. Application

4.7.1. Introduction

In this section, we apply the introduced techniques for analysis of development of stock market indicators.

Decisions on financial investment are usually based on a thorough analysis of the indicators for individual stocks (i.e., assets) and the outlook for the stock market. The main focus of investors is on the *return*, the *liquidity*, and the *risk*. The *return* measures the yield of the stock (dividend plus the change in the price of the stock relative to the price in the previous period) and the traded volume of the stock represents its liquidity. The value of the outstanding stocks, i.e., the size of the *market capitalization*, for investors indicates the importance of the company for the market developments. The *riskiness* of future returns is proxied by the volatility of the return (i.e., the variability of the expected return). It means that the higher the volatility, the higher the risk associated with the stock. Usually higher return is bound to higher risk, therefore the securities can be categorized as shown on the Figure 4.45. According to her risk profile, an investor is interested in particular asset categories. However, in general, stocks with higher returns keeping the volatility constant (at low levels) are preferred.

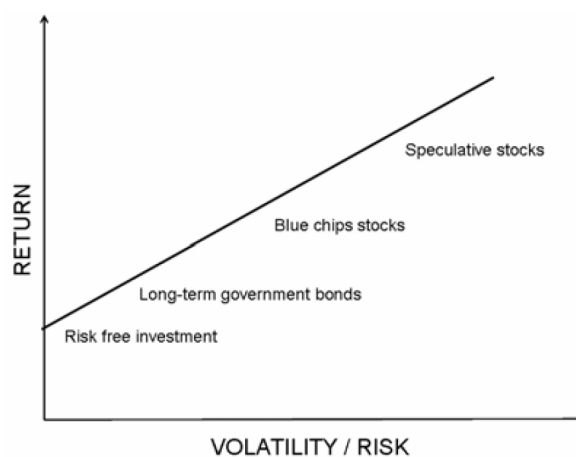


Figure 4.45.: Example of asset categories showing relationship between return and volatility [KV96].

The two indicators, risk and return, are time-dependent reflecting the activities on the stock market. Changes in risk and return lead to investment decisions and impact future performance of the portfolio (a set of stocks) that is currently held. In case the parameters change rapidly or the market is subject to turbulences, the composition of the portfolio needs to be adjusted in order to achieve the targeted returns and avoid large losses.

To support financial investment decisions based on *risk-return analysis* of financial items (stocks), the relationship between the two indicators is conventionally represented in the form of scatterplots. In this way, data are easily interpretable for financial analysts. The current systems used by the financial analysts, in general, do not allow for a simultaneous presentation of all these factors through time. Using the available techniques they can see a static picture of the relationship between risk and return but do not see the dynamics of the relationship between the stocks in their portfolio. Alternatively, they could display the development of either risk or return over time (i.e., as a line chart) for selected stocks but then it will not be possible to assess the correlation between the two factors.

In the following sections, we show how analysis of risk-return indicators for assets over time can be supported by the presented visual analysis tools. The use case for the visual analysis system presented in this section is twofold: to facilitate both the monitoring of financial market developments over time (individual assets and asset groups) and the analysis of the co-movement of stock market indicators for individual stocks and for asset groups (e.g., country grouping)). We concentrate thereby on the three types of tasks (as introduced in Section 3.1.1) involved in typical analytical processes. Firstly, we focus on visual exploration of the dataset using interactive animation and trajectory visualization. In this context we present results of a qualitative user study on the visualization tool (see Section 4.7.3). We then focus on asset groups' dynamics when analyzing country developments (see Section 4.7.4). Finally, we show how SOM clustering of trajectories of asset risk-return data can be used for revealing weekly patterns in dynamics of risk-return indicators (see Section 4.7.5).

4.7.2. Data

In this section, time-varying data on asset volatility and return are used. The data create a 2-D risk-return space $(\sigma, \pi)_i^k$, where (σ, π) refers to pairwise measures of risk σ and return π , observed for each financial asset k from a set of assets K : $k \in K$, and for each time stamp $i \in [0, n]$ of the observation horizon. Linear interpolation between all consecutive observations $(\sigma, \pi)_i^k$ and $(\sigma, \pi)_{i+1}^k$ results in risk-return movements (or trajectories) for the individual assets (see Figure 4.46 for an illustration).

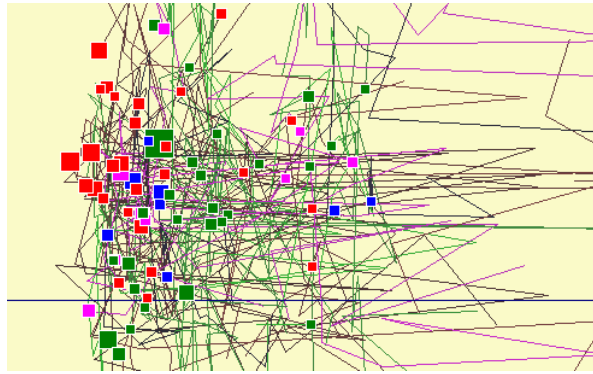


Figure 4.46.: Trajectories of risk-return data for a longer time period.

We consider two datasets. Both datasets contain data on daily risk and return for a set of assets over a period of time. In addition, also information on country of origin and market capitalization was provided. The country information is used for asset grouping. The analytical tasks and analytical process presented in Section 4.1.1 can be applied to these two real-world data sets.

The *first data set* contains data on 83 blue chips European stocks recorded between May 2005 and November 2006. It includes assets from four countries (Germany (30), Great Britain (33), France (9) and The Netherlands (11)).

The *second data set* considers 30 blue chip stocks listed in the Deutsche Aktien Index (German Stock Index) DAX [Deu]. The full data set spans a time frame between June 2005 and August 2007. The second data set contains assets only from one country (Germany), however it covers periods of March and August 2007 characterized by transient market turbulences.

The summary statistics on both datasets for the full time period are provided in the Figures 4.47 and 4.48. They show that the average return is around zero and volatility around 25-30. However the minima and maxima vary between countries and data sets. This difference is caused mainly by different time periods and asset composition of the data.

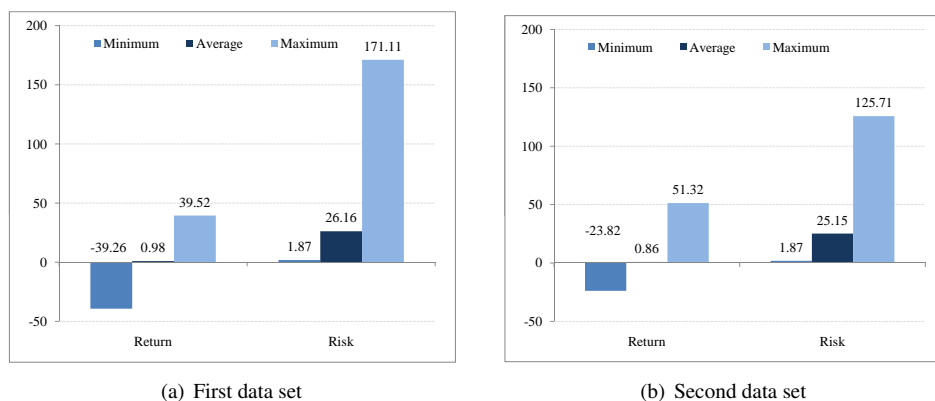


Figure 4.47.: Statistics on the first and second data set.

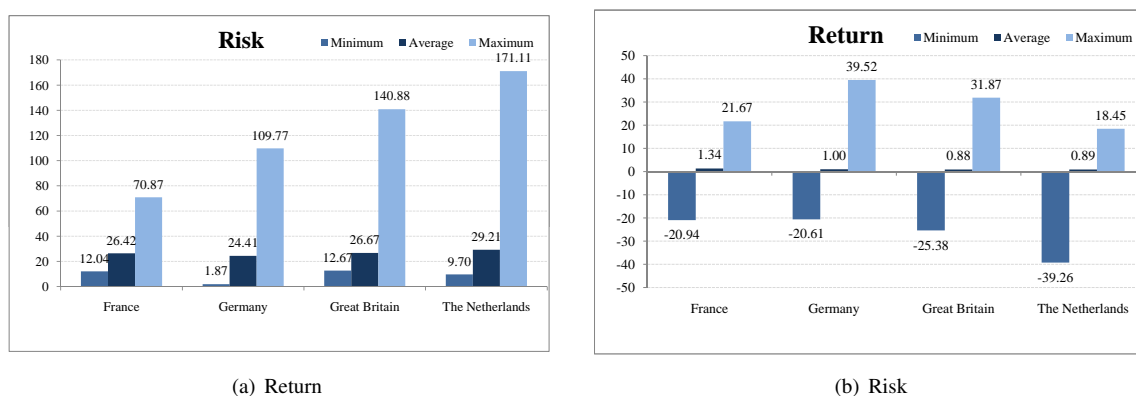


Figure 4.48.: Statistics on the first data set by country. Left: return, right: risk.

4.7.3. Visual Exploration of Time-Dependent Risk-Return Data

In this section, we concentrate on the visual exploration of dynamics in the stock market data based on the interactive scatterplot-based visualization tools described in Section 4.4. In our use case, as usually presented in financial applications, the x-axis in the scatterplot shows the volatility and the y-axis the return of the stocks. Further data on stock market capitalization, i.e., the importance of the companies stock for market developments, is encoded in the area of the rectangular glyph. The size of glyphs is normalized by minimum and maximum stock market capitalization in the sample. The country of asset origin is represented by the color of the glyphs.

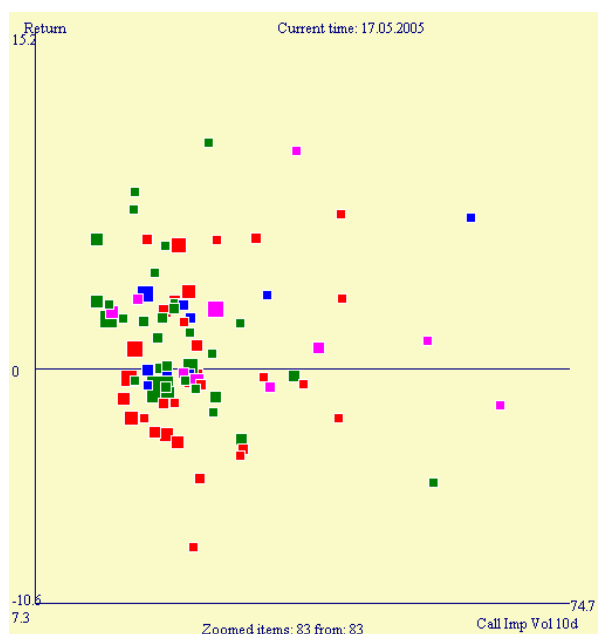


Figure 4.49.: The snapshot view on the risk-return stock market data. Color indicates country of origin and size the market capitalization.

For example, German stocks are colored red and French stocks are in blue (see Figure 4.49). It enables users to compare the development of the two stock categories more easily.

In the following, we show results of our observations of the data using the system presented and then present results of a qualitative user study of the system.

4.7.3.1. Experimental Results

Results for the first data set The speed of motion in *animated view* (reflecting the rate of change in the data) shows which financial instruments exhibit larger price movements. Using animation, we could spot in the data differences in size of movements between small and large assets (companies with small and big market capitalization). The movements of smaller assets were larger than those of large companies. This is in accordance with financial theory. Large companies tend to be more stable therefore the dynamics of their indicators should show lower volatility than of small companies.

In addition, when looking more closely on the dynamics of three German car producers during one week in June 2005, *trajectory* visualization (see Figure 4.50) reveals (dis)similarities between their behavior. A priori, the user might suppose that they move in sync. The trajectory feature however shows that the movements in return and risk of Volkswagen (VOW) and BMW (BMW) were indeed similar, whereas the evolution for DaimlerChrysler (DCX) was different.

Results for the second data set When analyzing the second data set, we compared the trajectories of the data before and during the turbulent weeks starting on February 26, 2007 (see Figure 4.51). The figure shows that

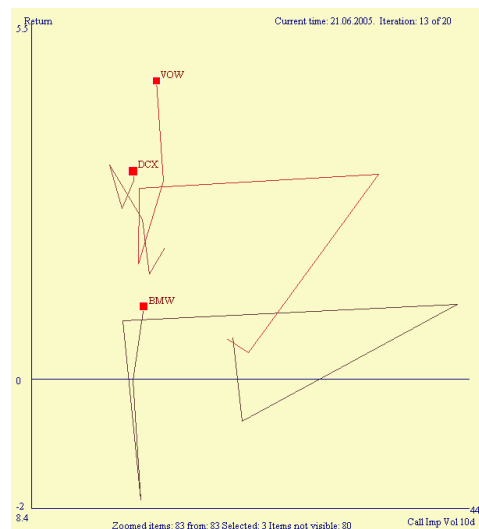


Figure 4.50.: Asset movements of three selected assets from the automobile industry (Volkswagen, Daimler and BMW) during one week in June 2005 showing similar movements of only two assets during this time period although all three assets are from the same industry. This indicates industry independent data movements of the third asset.

the movements in the first two thirds in February were much smaller (in particular with respect to the volatility changes) than in the week of 26th February. Moreover, the week of 12th March 2007 shows strong co-movements in the horizontal directions at the whole market meaning big changes in riskiness of assets owing to “current environment of market uncertainty” [Reu07]. These views confirm strong market movements in the respective periods.

4.7.3.2. User Feedback Results

We have conducted a small qualitative usability survey both with experienced users from the financial industry and with users that are not familiar with financial analysis. This feedback was used in early stages of the research as an input for further improvement of the system and opened new research questions.

Ten financial specialists and several persons not working in the financial industry were asked to answer a questionnaire about design and usability of the system. They were provided with the system for free testing. The experienced users also answered questions on the value added by the new software in their analytical work and possible usage in their business areas. Users also provided input for further functionalities of the system.

The *animation feature* was very positively ranked. The experienced users claimed it was very useful for their analysis. The comparison with other tools used in financial analysis shows that the animation is an interesting new feature which cannot be found in the currently used systems. In the analysis process using available tools, the users are forced to examine the data either across time or across entities, but not both at the same time. This poses major limitations for the analysis process and makes it cumbersome. The animation offers the possibility to better interpret market parameter changes of individual entities over time. The users would have appreciated the possibility of showing labels also during animation which was not available during the tests.

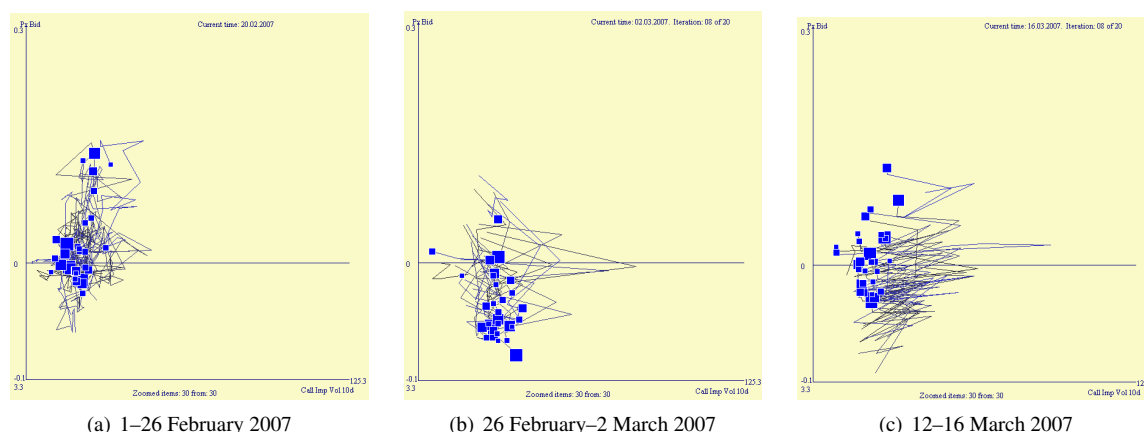


Figure 4.51.: Asset movements during selected time periods before and during the turbulent weeks at the end of February 2007. Left and center: Smaller independent movements during “normal” periods. Right: Large common movements during the turbulent period.

The *trajectory feature* helps analyzing the development of indicators of individual stocks during time also on static picture. However, some users found this feature confusing when too many stocks are shown, as it can be difficult to follow individual trajectories. For overcoming this problem, we later introduced clustering analysis of trajectory patterns (see Section 4.6).

The *filtering features* have proven to be very helpful. The users especially liked the threshold selection feature as it helps them easily identify stocks which are “good” or “bad” (i.e., low volatility and high return vs. high volatility and low return) and to observe their behavior over a longer period of time. The size threshold helps them to compare the data performance of small and big companies. Upon user request we also added the stock symbol search so that users can easily highlight stocks which interest them. Zoom and details on demand are standard tools and have proven necessary for the users in order to view more detailed information on the current state of the market.

In general, animation is good for understanding general movements in the data, however it is difficult to follow exact movements of individual items over longer time periods. For showing exact movements, trajectories are more suitable. The tool was overall deemed useful for decision making support and analysis of market movements.

4.7.4. Visual Analysis of Time-Dependent Risk-Return Data With Asset Grouping

In this section, we show how the monitoring of specific features for data entities within groups, entire groups and inter-group relationships can help the analysis of risk-return data over time. Please note that in the previous subsection, this aspect was largely neglected.

In financial analysis, individual assets are often grouped according to their sectoral, country or portfolio membership. In our application example, we use country groupings. In such a grouping, the dynamics of the individual assets are compared in relation to the overall country development, or countries’ dynamics are compared. In order to illustrate this task, a data set for German assets is used. In a next step, the dynamics of individual countries are analyzed relative to each other. For this task, we use the first dataset with assets from four countries. The

examples presented, show that the combination of these indicators can bring to the fore extraordinary movements in financial markets.

4.7.4.1. Results for Visual Analysis of Asset Dynamics within a Country

The analysis of individual assets in a group concentrates on extraordinary dynamics of the asset, the asset position in a group (e.g., to the country) etc. For example, it is interesting whether the asset co-moves with the group, whether it moves more or less rapidly than the group, whether its risk-return profile (i.e., position) moves in the same direction as the group etc.

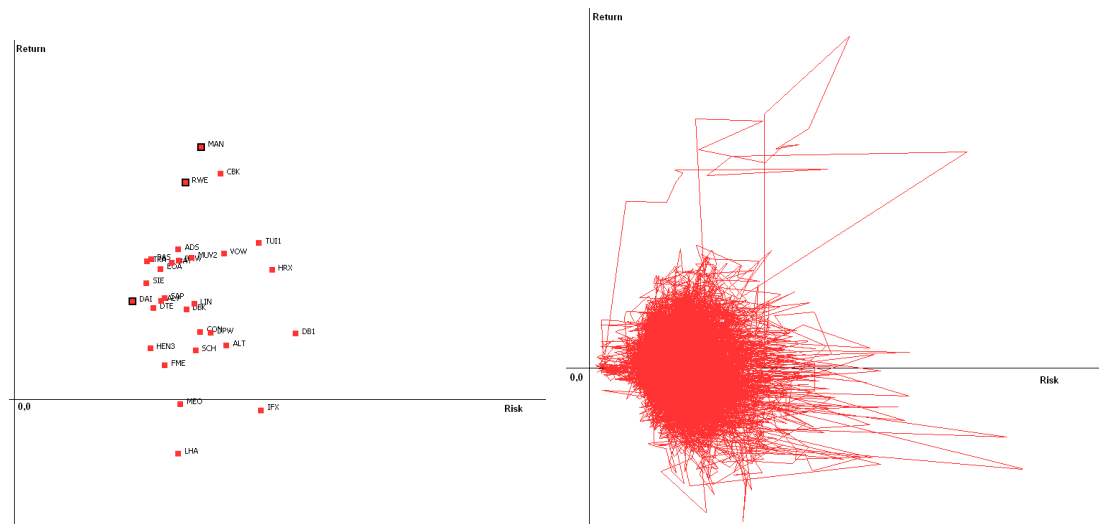


Figure 4.52.: Visualization of the individual asset data from the first data set. Left: The individual assets from DAX data set at the start of the analyzed period. Assets MAN, DAI and RWE (on the left side of the group) are selected for closer inspection. Right: The strongly overplotted trajectories of individual assets during the whole examined period showing the need for analytical monitoring of the data.

In our analysis we selected three assets: (MAN (MAN AG, ID=20), DAI (Daimler AG, ID=10) and RWE (RWE AG, ID=23). These stocks are situated on the left border of the group at the start of the analyzed period, meaning they have very good risk-return profile having minimum risk at the given return level (see Figure 4.52 left). We use the monitoring of entity features for assessing the dynamics of the three assets in the group.

The visualization of asset trajectories in the risk-return plot does not reveal any particular insights into the data (see Figure 4.52 right). However the visual analysis of asset features over time (see Figure 4.53) shows that there was strong movement of assets at the end of the period (according to step length and distance to mid-point). Although the stocks have very volatile movements (according to the movement direction) they broadly co-move with the group (according to the small length and direction difference to the mid point).

In these views, two dates however stand out: 16. August 2007 and 19. September 2006. On both dates, one of the monitored assets moved far from the average stock market point. By closer look at 16. August 2007, we can see that the distance to the group mid point is very large especially for Daimler AG asset (see Figure 4.54). In general, on this date, the stocks are spread on a large area owing to the strong market turbulence after breaking out of the subprime mortgage crisis (see also analysis of group dynamics in the following paragraph). The view

on the stock market on the second data indicated by the monitoring of the features, 19. September 2006, reveals that MAN AG is “far” from the rest of the group of assets (see Figure 4.55). This phenomenon is caused by the news of possible acquisition of Scania by MAN AG [MAN06].

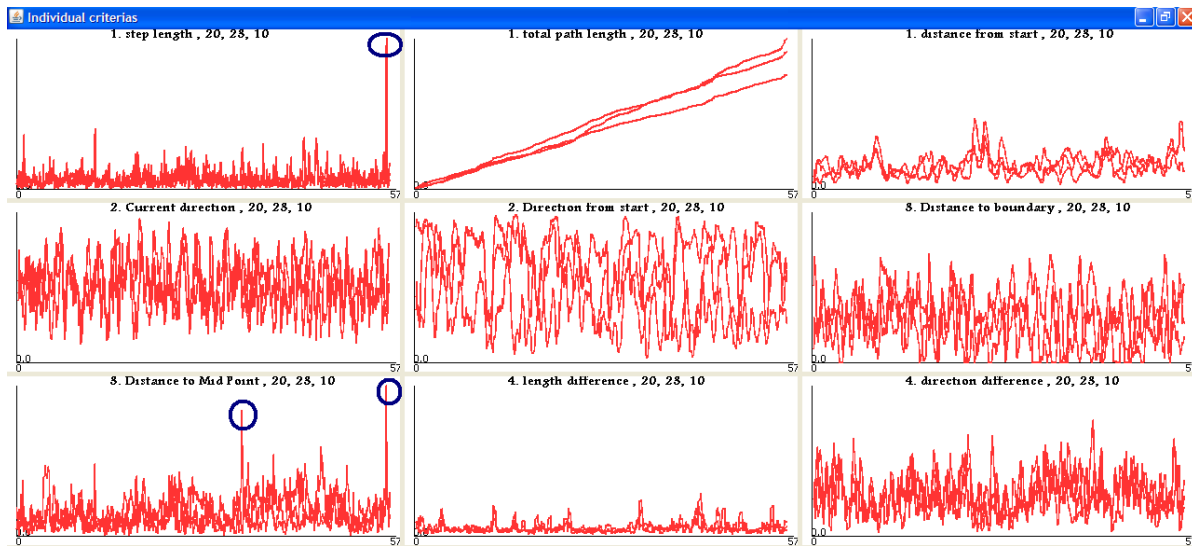


Figure 4.53.: Monitoring of the time-dependent features for the three selected assets (MAN, DAI and RWE). Exceptional movements owing to individual market shocks are marked with blue circles.

4. Visual Analysis of Two-Dimensional Time-Dependent Data

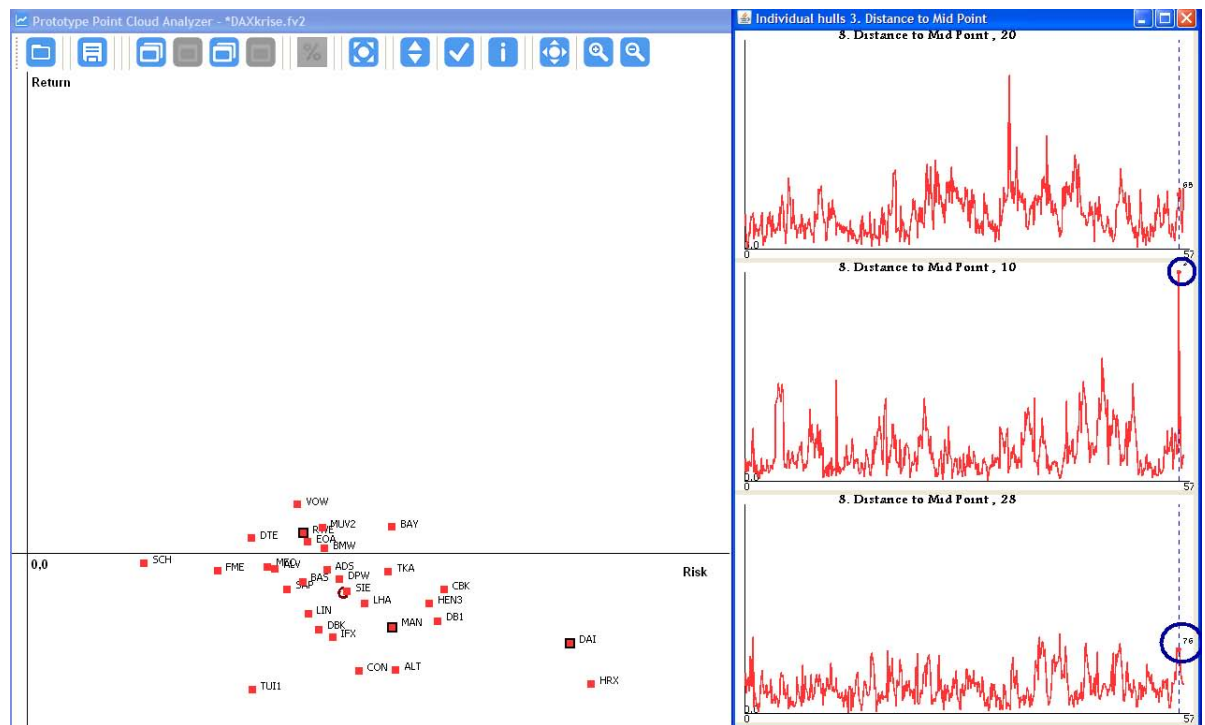


Figure 4.54.: Extraordinary movement of selected stocks on 16. August 2007 caused by the start of the subprime mortgage crisis. Left: Snapshot of the data on the selected date. Right: The feature monitoring with highlighted selected time point (blue dashed line) based on the distance to mid point feature.

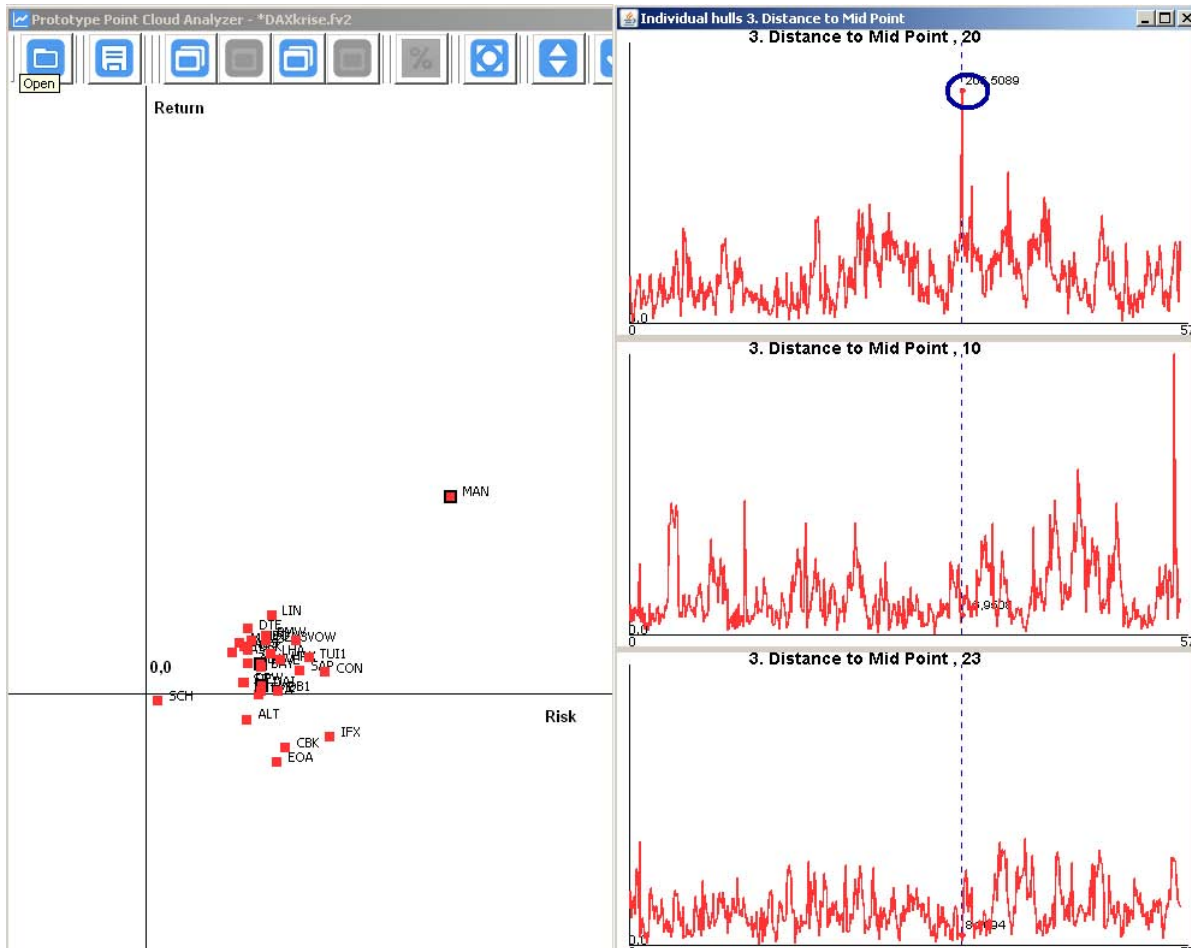


Figure 4.55.: Extraordinary movement of MAN AG on 19. September 2006 determined by its potential acquisition of Scania. Left: Snapshot of the data on the selected date with MAN highlighted in the right upper corner. Right: The feature monitoring with highlighted selected time point (blue dashed line) based on the distance to mid point feature.

4.7.4.2. Results for Visual Analysis of Country Dynamics

For the visual analysis of groups of two-dimensional time dependent data, in our case of country developments, two types of visual abstractions are used: hulls and mid-points. There are several hull shapes that can be applied (e.g., convex, alpha, PCA-based, rectangle etc.). When analyzing stock market data, from the variety of these hull-based abstractions, convex hulls are most appropriate. Convex hulls show all possible assets that can be created by combining multiple assets within the group.

Alternatively, the mid point abstraction can be applied. The mid-points (the gravity center of the group) are in this case an approximation of the so-called market portfolio (a portfolio built as a sum of assets on the market). Thereby they form also a suitable abstraction for the group of assets.

We use both abstractions in the following analysis of the second data set consisting only of German stocks.

The visualization of the group of German stocks together with their mid-point at the beginning of the analyzed period is shown in the Figure 4.56 left. The figure on the right shows the development of the group of assets over the whole time period. Owing to the strong overplotting, this view is however difficult to interpret. Therefore we calculate and visualize moving averages of group features for German stocks (see Figure 4.57). The size features (top left) indicate several time periods of higher spread of the assets, however density indicators (top right) remain relatively stable. Also mid point distance from boundary and geometric hull center are stable. Therefore, no extraordinary movements are easily spotted from these two types of indicators. In contrast, shape indicators (in particular convexity and relative PCA length) indicate extraordinary dynamics on three dates which we closely examine in the following.

When looking at the three selected dates: 22. March 2006, 8. May 2007 and 16. August 2007 (see Figures 4.58, 4.59 and 4.60), one phenomenon becomes apparent.

The first two dates show a high density of assets in a small area and an outlier. This outlier causes large area and diameter with high values of outlier indicator, while convexity is at low levels. The outlier in the first case is Schering showing extraordinary movements caused by its potential acquisition by Merck AG at the beginning of March 2006 [Fra06]. The second outlier is Altana AG caused by speculations about extremely large dividend yield to be paid out after an outstanding result in the first quarter of 2006 [Alt06a] and [Alt06b].

The third example was selected because of the maximum area indicator. In contrast to the previous two cases, the high area, high diameter and lower density are not connected with high levels of outlying indicator. Indeed, the relative length of PCA eigenvalues is small and sparsity high. Interesting are also the indicators of the mid point distance from start and mid point velocity (step length) at their highest levels. This confirms strong movements on the whole market in this period owing mainly to uncertainties in the initial phase of the subprime mortgage crisis (see Figure 4.61 for illustration of the movement). In contrast, in the first two cases, the high(low) values of the features were caused by strong movements of individual assets (Schering AG and Altana AG).

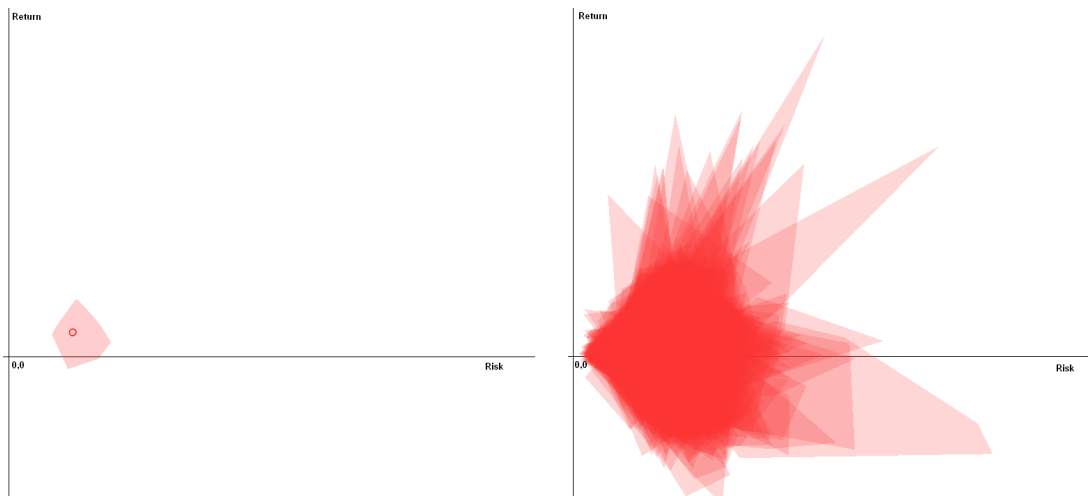


Figure 4.56.: The visualization of the group of German stocks. Left: The data at the start of the period showing compact state of the market. Right: Tracks of the developments of the group over the whole period showing strong overplotting of the traces leading to the need for further feature-based analysis.

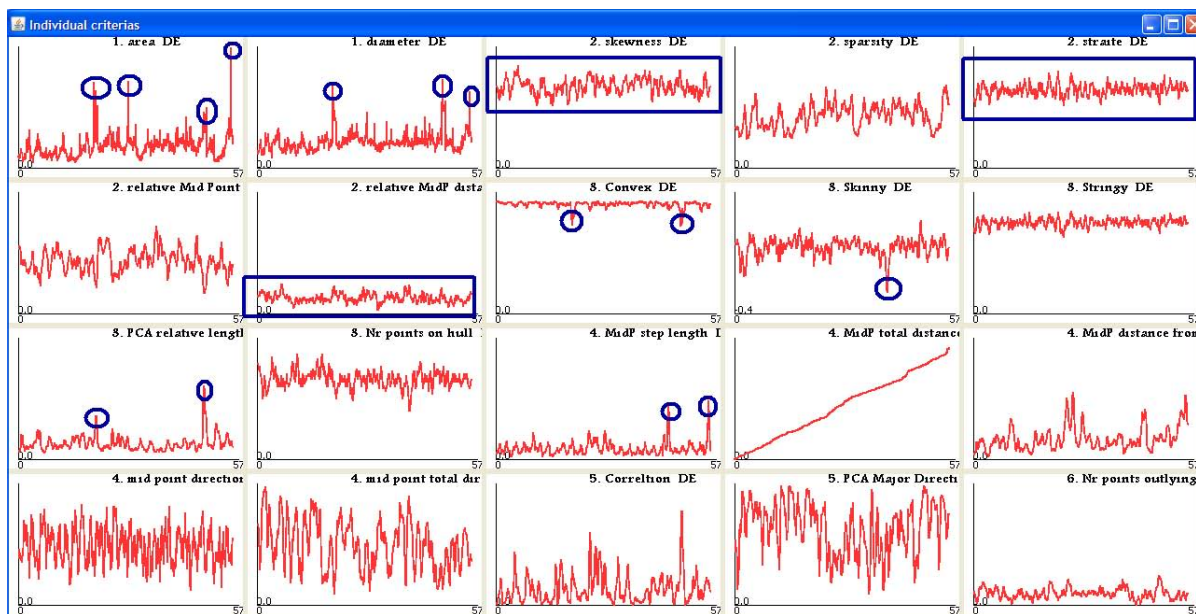


Figure 4.57.: The monitoring of developments on the German stock market using a set of features. For example, the area (group size) feature indicates several exceptional time points which are analyzed in more detail.

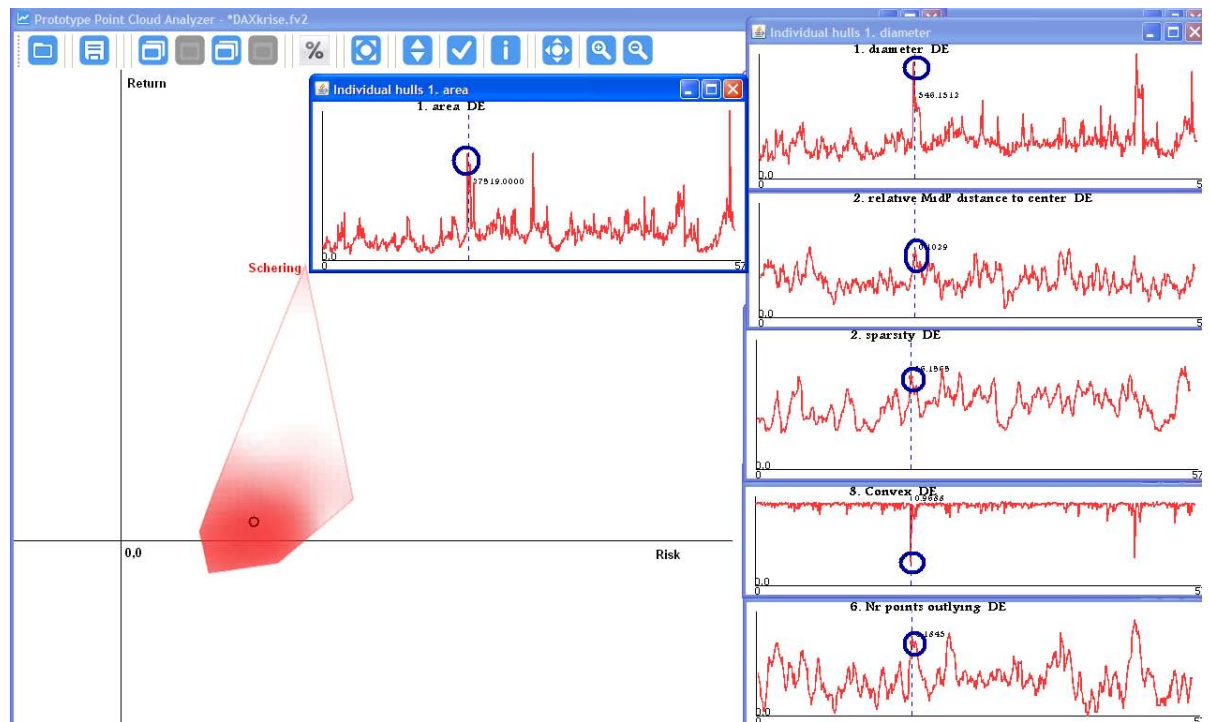


Figure 4.58.: The monitoring of developments on the German stock market on 22. March 2006. The selected time point is indicated by the blue dashed line. It shows extraordinary movements of Schering asset caused by its potential acquisition by Merck AG. The Schering asset was an outlier on the market. This exceptional movement is identified by several features (e.g., area, convexity, diameter, relative mid-point distance or sparsity). The selected time point is indicated by the blue dashed line.

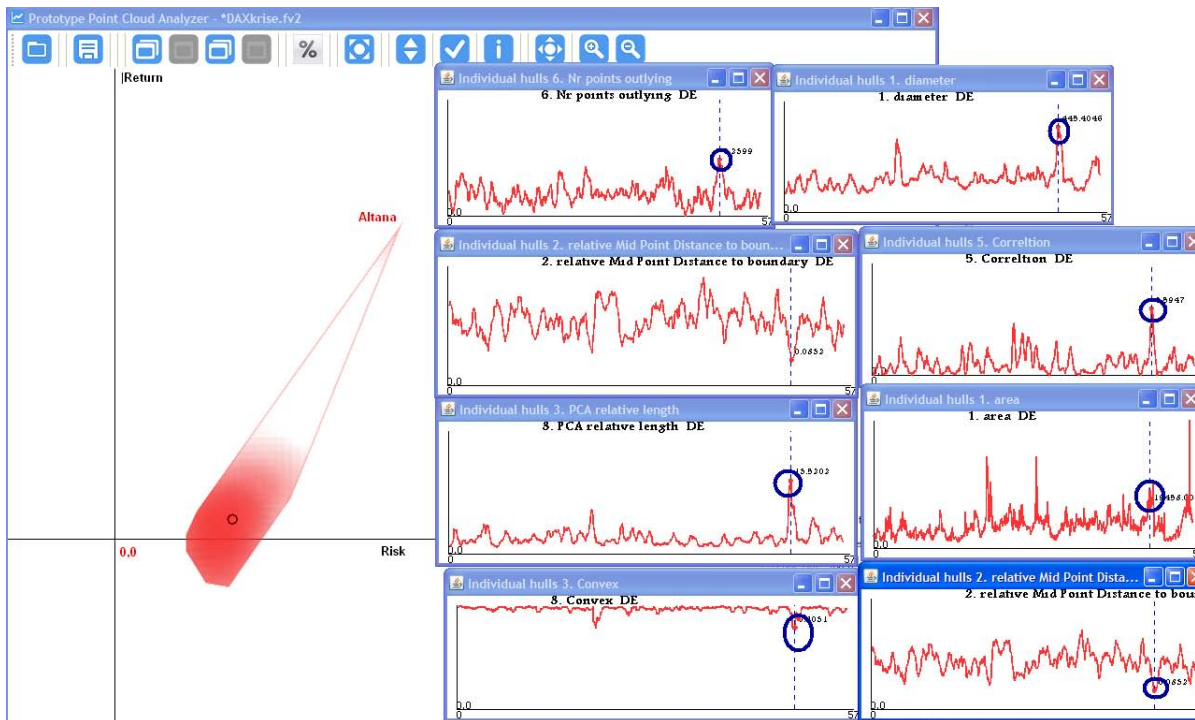


Figure 4.59.: The monitoring of developments on the German stock market on 8. May 2007 revealing Altana outlier asset. This situation is caused by speculations about future financial status of the company. The selected time point is indicated by the blue dashed line. Several features are used for identification of this event (e.g., PCA relative length, area, convexity, diameter, or relative mid-point distance to boundary).

4. Visual Analysis of Two-Dimensional Time-Dependent Data

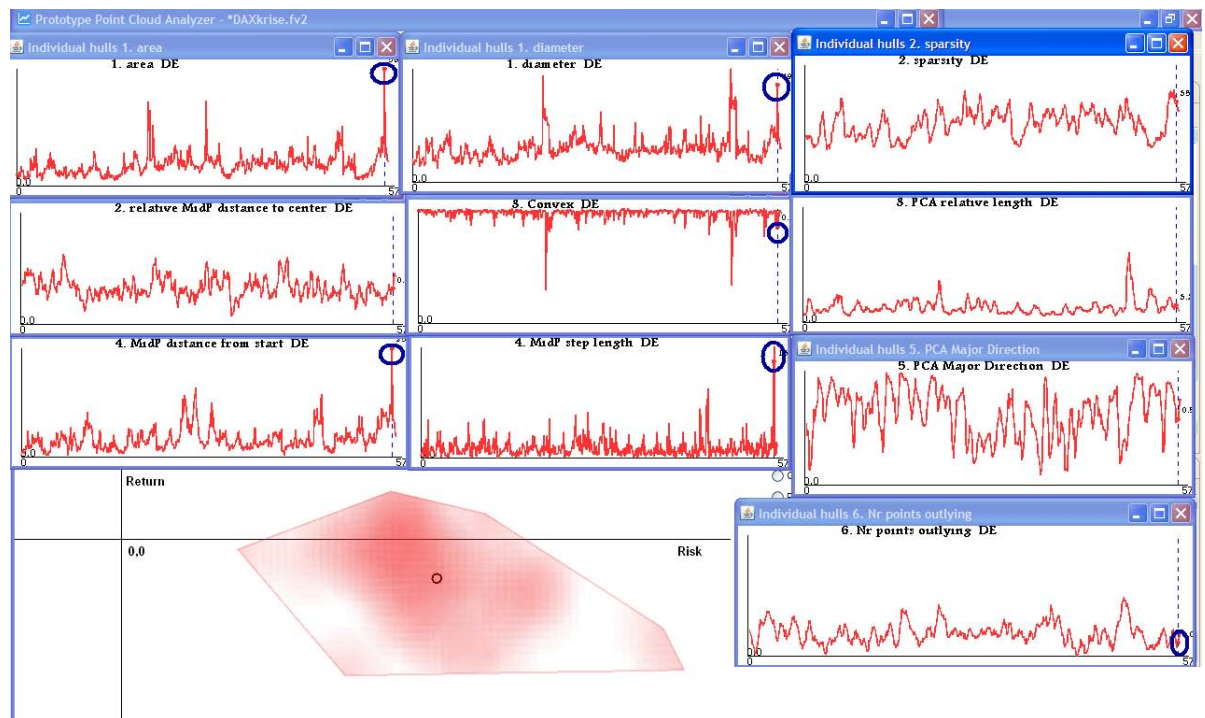


Figure 4.60.: The monitoring of developments on the German stock market on 16. August 2007 indicating the strong market movements caused by the outburst of the subprime crisis. The blue dashed line marks the event by several features such as area, diameter, convexity, mid-point distance from, start and mid-point step length.

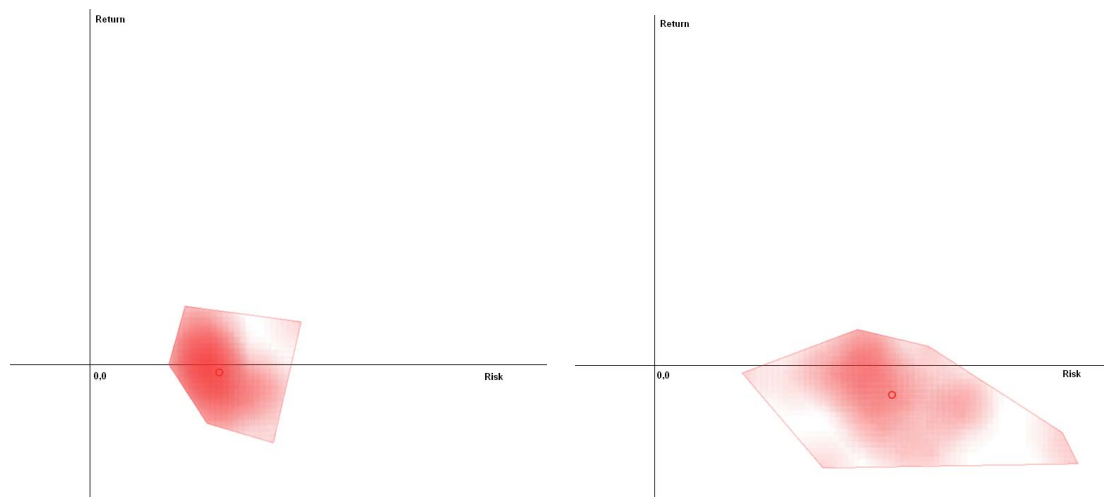


Figure 4.61.: The visualization of the German stock market on 15. and 16. August 2007 (left and right) showing rapid market movement caused by the outburst of the subprime crisis.

4.7.4.3. Results for Visual analysis of Inter-Country Dynamics

We now concentrate on the analysis of co-movements and inter-country positions for assets from four countries (LN (Great Britain), GY (Germany), FP (France) and NA (The Netherlands)). When looking at the positions of the countries' assets at the beginning of the period (see Figure 4.62 left) we can already see their strong overlap. This overlap endures in the whole time period (see Figure 4.62 right). However, in this view, we cannot examine the dynamics of the size of this overlap and also of the distances between the country assets. Therefore we use the monitoring tool (see Figure 4.63) for assessment of the developments in the four countries according to several criteria. The monitoring clearly shows that all countries overlap during the whole time period (see number of overlap hulls indicator) whereas the (relative) overlapping area and the distance between countries vary over time. In particular, the maximum distance between mid-points indicates a short time period of strong diversity (at the end of the second third of the time period). When looking closely on the data in the middle of this period (see Figure 4.64a) and compare it with a date with small distance (see Figure 4.64b) it stands out that stocks from the Netherlands have much higher risk (are positioned far right) than the other countries, which is not prominent on 10. October 2006. It indicates country specific movements in The Netherlands in May 2006.

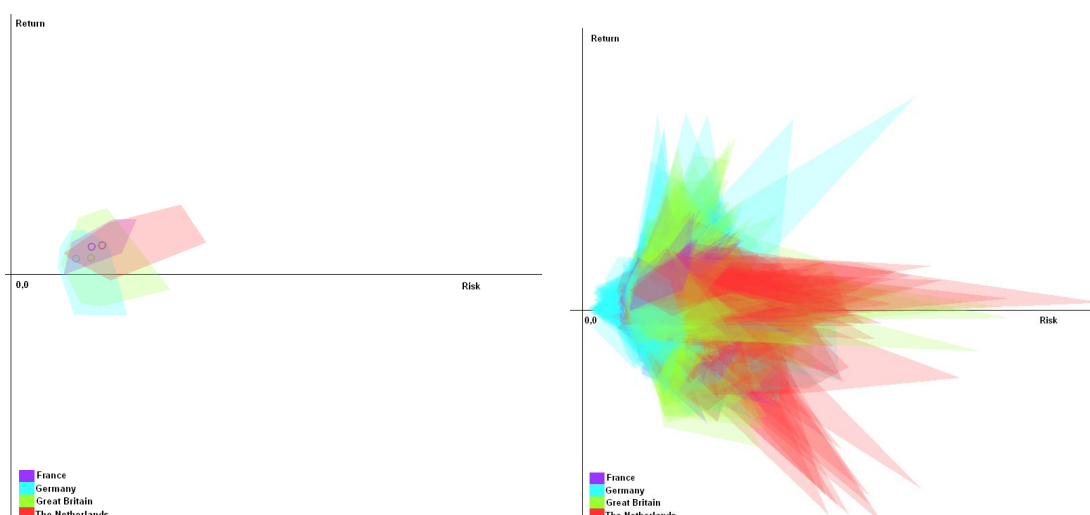


Figure 4.62.: The visualization of the country groups. Left: The data at the start of the period showing the overlap of the four country groups. Right: Tracks of the countries' developments over the whole period showing strong overplotting thereby the need for further analysis using feature monitoring.

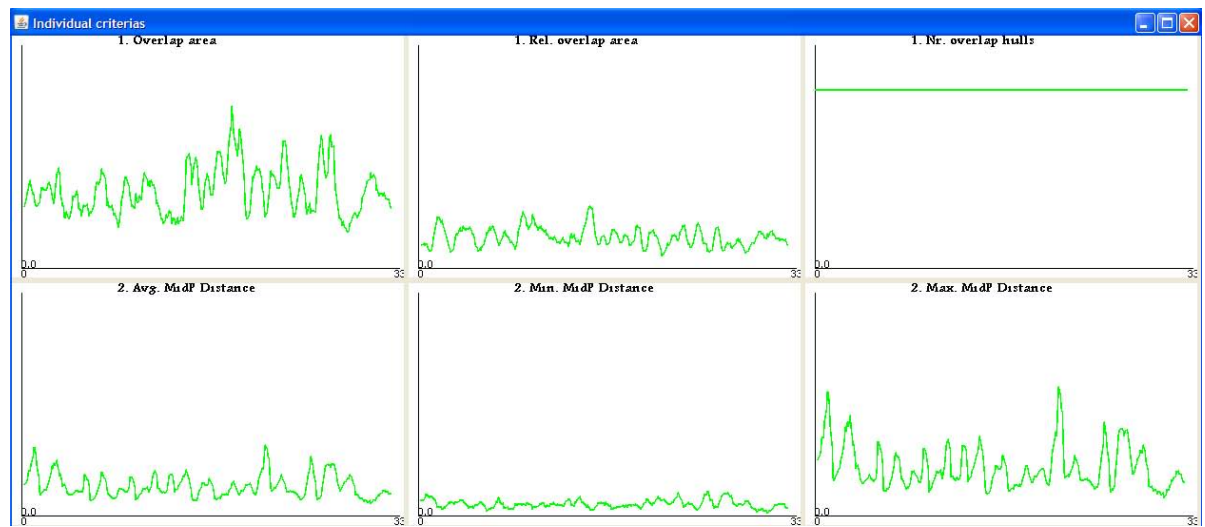


Figure 4.63.: The monitoring of inter-country developments on the stock market using group relationship features over time. The monitoring indicates overlapping of all countries over the whole time period with varying overlap extent. Especially maximum mid point distance varies significantly which is analyzed in more detail.

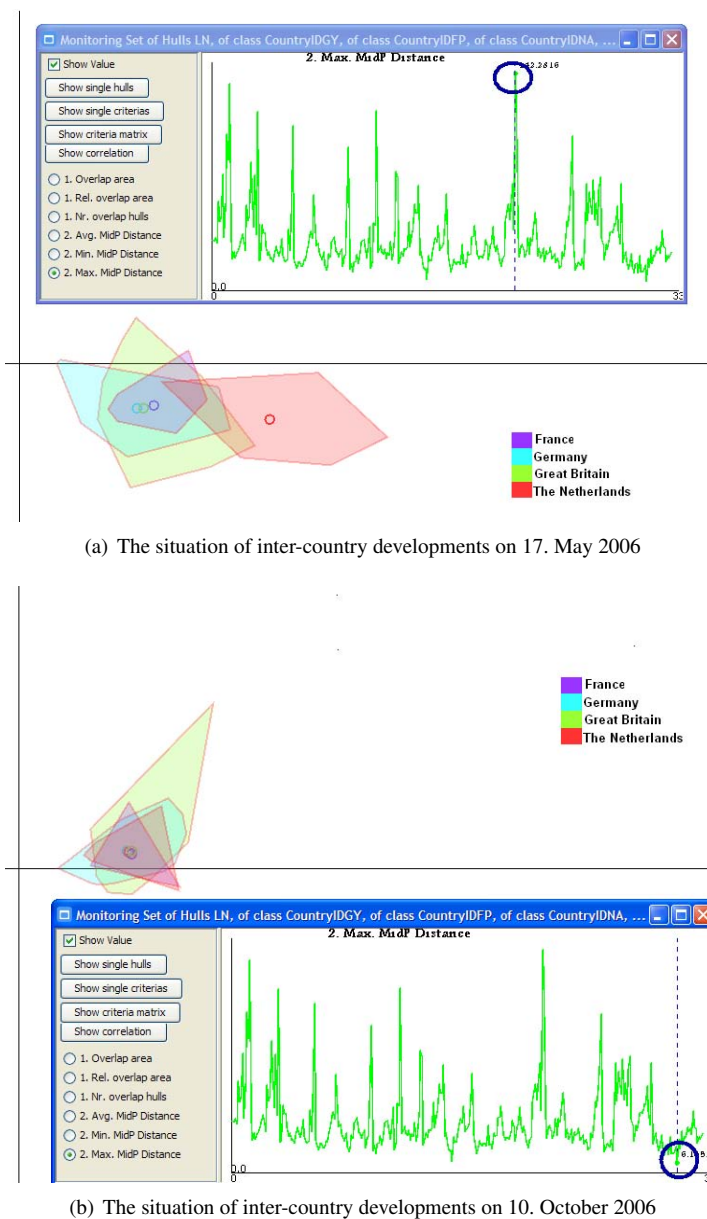


Figure 4.64.: The monitoring of inter-country developments on the stock market. a) on 17. May 2006 determined by high values of the maximum mid-point distance feature highlighted by blue dashed line in the chart. The view on the data in the main window confirms large differences in stock market situation in the Netherlands. b) on 10. October 2006 based on low levels of maximum mid-point distance (highlighted by blue dashed line in the monitoring window). Main view confirms this indication by strong overlap of all groups of assets.

4.7.5. Visual Analysis of Time-Dependent Risk-Return Data using SOM Clustering

In this section, we show how SOM-based clustering can be applied to the *pattern analysis of risk-return data*. We first explain the details on feature extraction and SOM parametrization for the daily risk-return data sets and then show clustering results for both data sets. Using visual interactive clustering, the common and extraordinary market movement patterns are revealed.

4.7.5.1. Set-up for Analysis of Risk-Return Data

Trajectory fragmentation: Before starting feature extraction, we partition each risk-return trajectory for the whole time period into suitable trajectory fragments (subsets). In the financial analysis domain, days, weeks, months are meaningful time granularities. Given the daily data sets for a period up to two years, weekly asset movements seems an appropriate choice. It allows us to study day of week effects, which is also supported by financial literature (see for example [Sch02, HK95, FRE80] for more information on seasonality in trading patterns). We therefore decompose the risk-return sequences of each asset k into fragments of 5 consecutive, daily observations representing full trading weeks (Monday to Friday), obtaining trajectories consisting of 5 points and 4 segments linearly interpolating the points. Let $T_{F_w}^k = [(\sigma, \pi)_{\tau(w)}^k, \dots, (\sigma, \pi)_{\tau(w)+4}^k]$ denote the sequence of daily risk-return observations for asset k during week w , where $w \in W$ is an index over the set of weeks W contained in the observation horizon, and $\tau(w) : w \rightarrow t$ is a function yielding the time stamp of Monday in week w .

In this set-up, the task of the clustering algorithm is to reduce the (possibly large) set of weekly risk-return sequences $T = T_{F_w}^k | k \in K, w \in W$ to a smaller set of s prototype trajectories $P = \{p_1, \dots, p_s\}$, where s is given by the user, or determined by cluster validity analysis.

Decomposing the first data set into full trading weeks yielded 66 weeks of observations, totaling 5478 chart movement (trajectory) samples as input to the SOM-based cluster analysis. The second data set contains 105 weeks of market movements (owing to the longer whole examined time period) of 30 assets creating 3150 chart movements.

Feature extraction and normalization Firstly, we represent each weekly trajectory segment $T_{F_w}^k$ by a simple 10-dimensional feature vector obtained by concatenating its sequence of normalized (σ, π) (geometric) coordinates. Then, the normalization is done by linearly scaling each sequence to span the interval $[0, 1]^2$ (local min-max normalization). This representation implies invariance with regard to position and scale, but not with regard to rotation or more complex geometric transformations. This allows for analysis of pattern shapes irrespective of the size of the movements and the risk-return position of the asset. Please note that if position or size of the movements were interesting for the analytical task, another normalization (e.g. global min-max) can be used.

In addition to the geometric features, we also calculate abstract features measuring the size of the movement (showing the asset stability), the direction of the movement (showing whether the asset position changes towards more riskier/higher return or lower riskier/lower return assets), changes in direction during the week (showing stability of the movement direction), etc..

SOM Parameters We trained a Self-Organizing map consisting of 12×9 prototype vectors arranged on a grid of rectangular topology. Other grid sizes can be used. We here use about 100 chart movement prototypes as a compromise between accuracy of the representation, and level of abstraction achieved. The SOM learning

algorithm was configured using standard parameter settings as suggested in [KHKL96]. Owing to the topology-preserving properties of the SOM, neighboring prototypes show similar patterns.

In the remainder of this section, we exploit the clustering results for both data sets at disposal described in Section 4.7.2. In the following, we concentrate on geometric features noting that these features have shown meaningful results for this particular application task. However, additional abstract features may be meaningful for other data sets or analytical foci.

4.7.5.2. Results for the first data set

The result of SOM training in unsupervised mode is shown in Figure 4.65. Inspection of the generated Self-Organizing Map indicates that a suitable, meaningful clustering result was obtained. The SOM nicely organizes the space of movement patterns by arranging prototype trajectories on the SOM grid such that neighboring patterns are similar to each other, and the different patterns are smoothly transitioning over the map.

Figure 4.66 shows four example movement patterns located at the corner areas of the SOM. Note that the patterns roughly represent the four possible diagonal movements in the risk-return chart space, e.g., pattern (a) represents a decline in both dimensions, while pattern (e) represents the opposite direction. These four pattern types reflect the most salient, discriminative chart movements possible, and serve as a good starting point for interpretation of the pattern distribution with regard to the given SOM grid. Owing to the topological properties of SOM maps, these four possible diagonal chart movements in the corner area of the map are accompanied with smoothly transiting patterns in between. The middle area shows more complex patterns, with circular or self-intersecting characteristics. In addition, the pattern fit to the actual data is shown by trajectory bundles (see Figure 4.42 on page 190), communicating the frequency of matched data samples and the variance around their associated prototypes.

When analyzing patterns of a selected asset (Schering AG) using the object-oriented view (see Figure 4.67) the following observations stand out. Although, the global view shows a high frequency of patterns of type (a) in Figure 4.66, these do not occur for the selected asset. Also, the asset does not show complex chart movement patterns located centrally on the prototype grid. Some of the most frequent patterns occur in the bottom-right area of the map, meaning that these patterns are typical for this asset. This is in line with our findings in the previous section (in particular during March 2006 caused by potential acquisition of Schering AG by Merck AG). Based on the asset view, a semi-automatic alert system can be set up, monitoring current chart patterns, notifying the analyst of possibly atypical patterns occurring in real time.

Additionally, an entropy analysis of pattern frequency considering the grid-based pattern distances finds the most prominent patterns in the weeks 46 and 48 (see Subsection 4.6.5.4 for details). These patterns represent those patterns in the identified weeks, where presumably, important global factors dominate the market dynamics. Two selected patterns ([6,6] and [0,2]) were highlighted in sequence view (see Figure 4.68).

The results show that the pattern highlighted in blue occurs most prominently in the 46th and 48th weeks, while the yellow highlighted pattern does so in week 45 and 50. The sorting of the asset rows by the first pattern in the two most frequent weeks reveals (a) a strong correlation between the blue pattern for weeks 46 and 48, and (b) correlations between the two patterns for subsets of the assets. This is an interesting finding, as the yellow pattern represents a decrease in both risk and return dimensions, while the blue pattern indicates an increase in risk combined with a moderate change in return. This insight encourages further investigation of the analyst, considering additional information sources such as a financial news archive for June 2006.

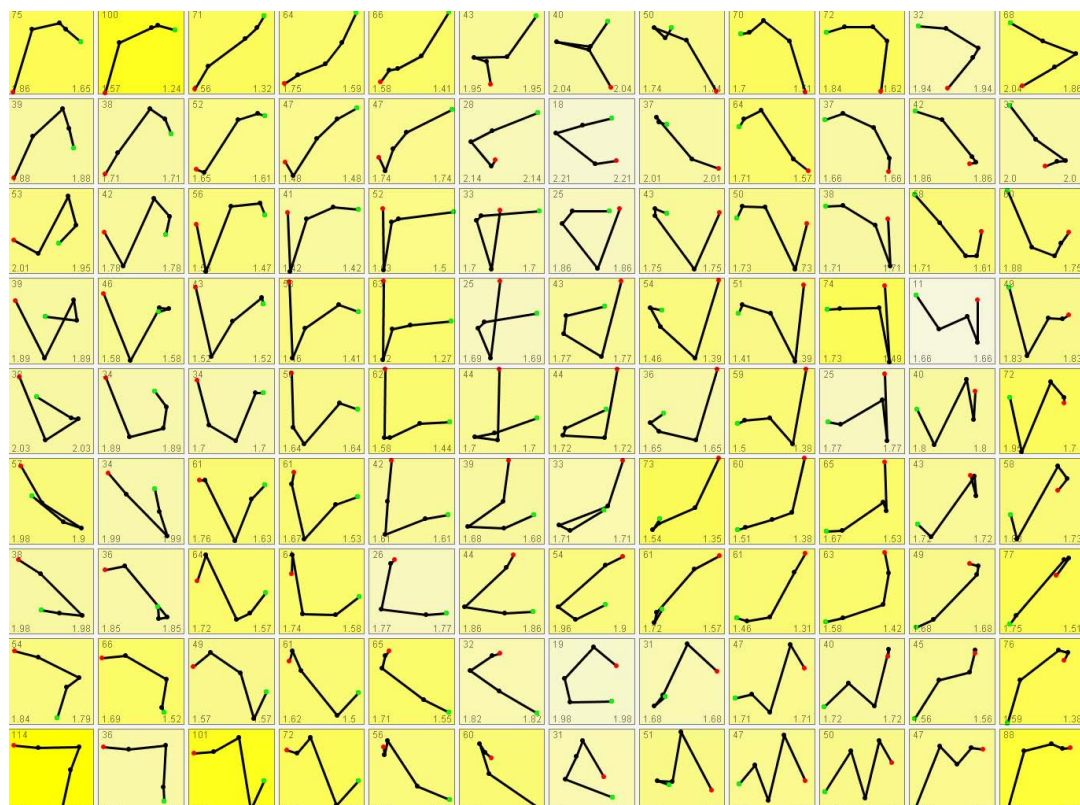


Figure 4.65.: Prototype patterns occurring in the first data set. The pattern frequency is indicated by the background color from white (low) to yellow (high). The view shows reasonable results smoothly transitioning across the SOM grid.

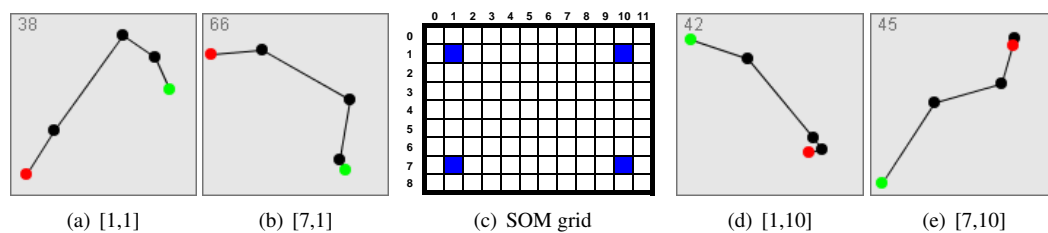


Figure 4.66.: Prototype patterns occurring at the corner areas of the SOM grid (coordinates indicate row and column). These patterns correspond to the four main risk-return movements: a) decreasing volatility and return, b) decreasing return with increasing volatility, d) increasing return and decreasing risk, and e) decreasing risk and return.

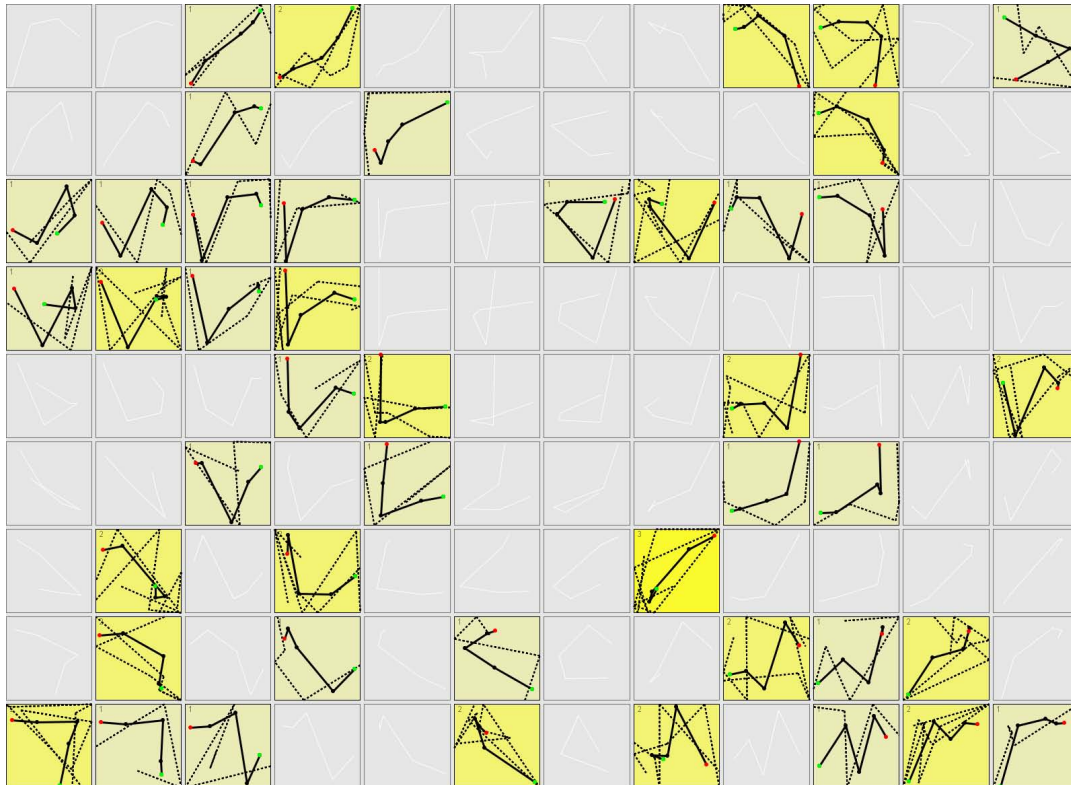


Figure 4.67.: SOM result restricted to patterns of the Schering AG asset. This view shows that the distribution of the Schering movement patterns does not correspond to general data patterns (see Figure 4.65).

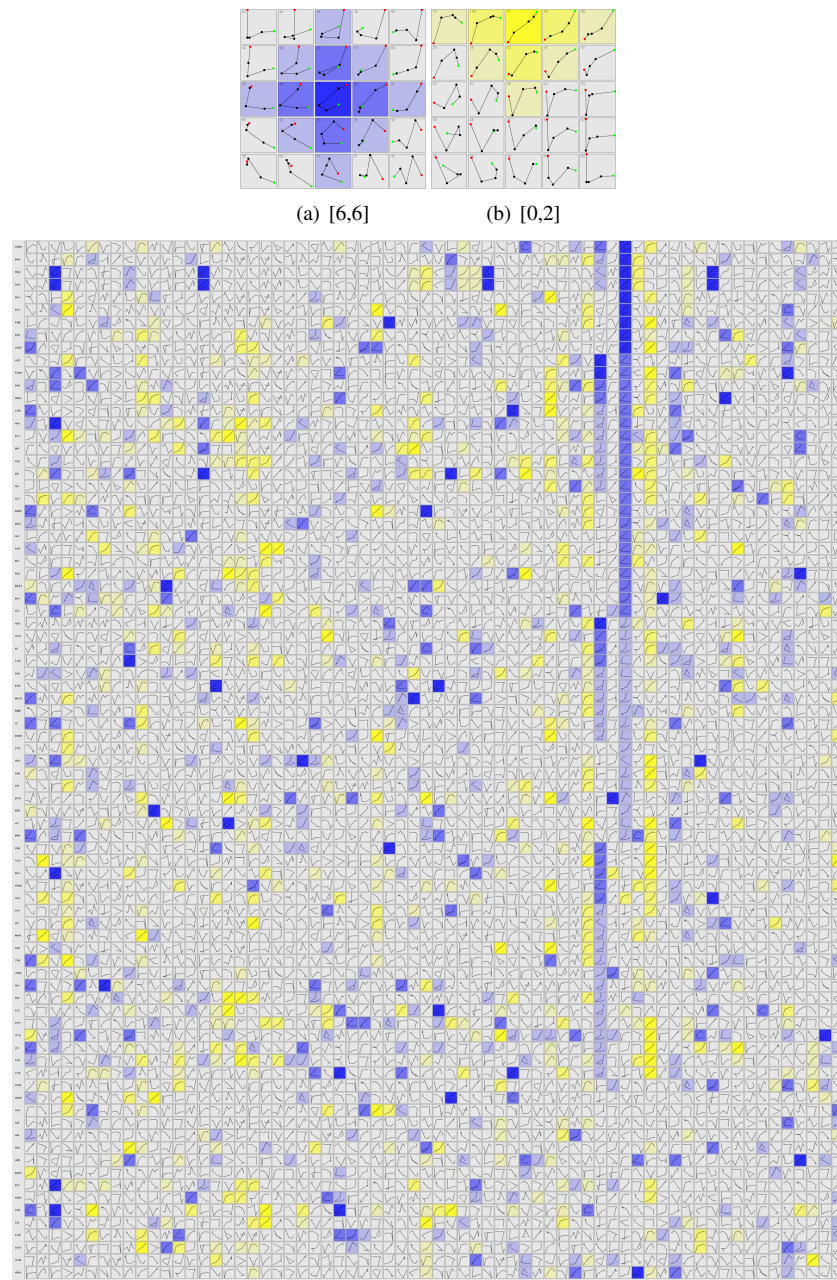


Figure 4.68.: The sequence view visualizing the chart movements of 83 assets during 66 weeks of observation. Patterns [6,6] and [0,2] as selected by the user from a set of automatically generated candidate patterns (top) are highlighted in the sequence view (bottom).

4.7.5.3. Results for the second data set

The result of SOM training in unsupervised mode for the second data set is shown in Figure 4.69(a). Yellow color-coding shows the relative density of matched sample charts over the Self-Organizing Map. It can be seen that the distribution of the patterns in the dataset is relatively uniform, meaning that all the found patterns occur with similar frequency during the whole time period. The shapes of the patterns vary substantially and cover the important types of market movements being similar to the patterns of the first data set.

In the following, we look closely at the market movements during the first three weeks of March 2007, when a transient market downturn leading to significant drop of many of the listed stocks' prices occurred. Figures 4.69(b) to (d) indicate the patterns occurring during these weeks. The density of matched samples, as well as their spread (deviation) from the respective cluster prototypes is indicated by background highlighting (yellow) and trajectory bundles (blue). In contrast to the whole time period, the pattern for the turbulent weeks show that the distribution of patterns changes drastically. The variance of the market movements seen during normal trading weeks is replaced by strong developments in one direction on the whole market. The trading week of February 26–March 02 (Figure 4.69(b)) first shows an increase in daily stock price return (y-axis, upward movement), while showing increased risk (price volatility) at the same time (x-axis, rightward movement) for most of the traded stocks. Followed by this upturn, a downturn was observed for many stocks, as characterized by a decrease in daily return (downward movement along y-axis) together with fluctuations in variance (movements along x-axis). The downturn is dominating the risk-return chart patterns occurring in the latter two weeks (Figures 4.69(c) and (d)). These findings confirm the results shown in Section 4.7.4.

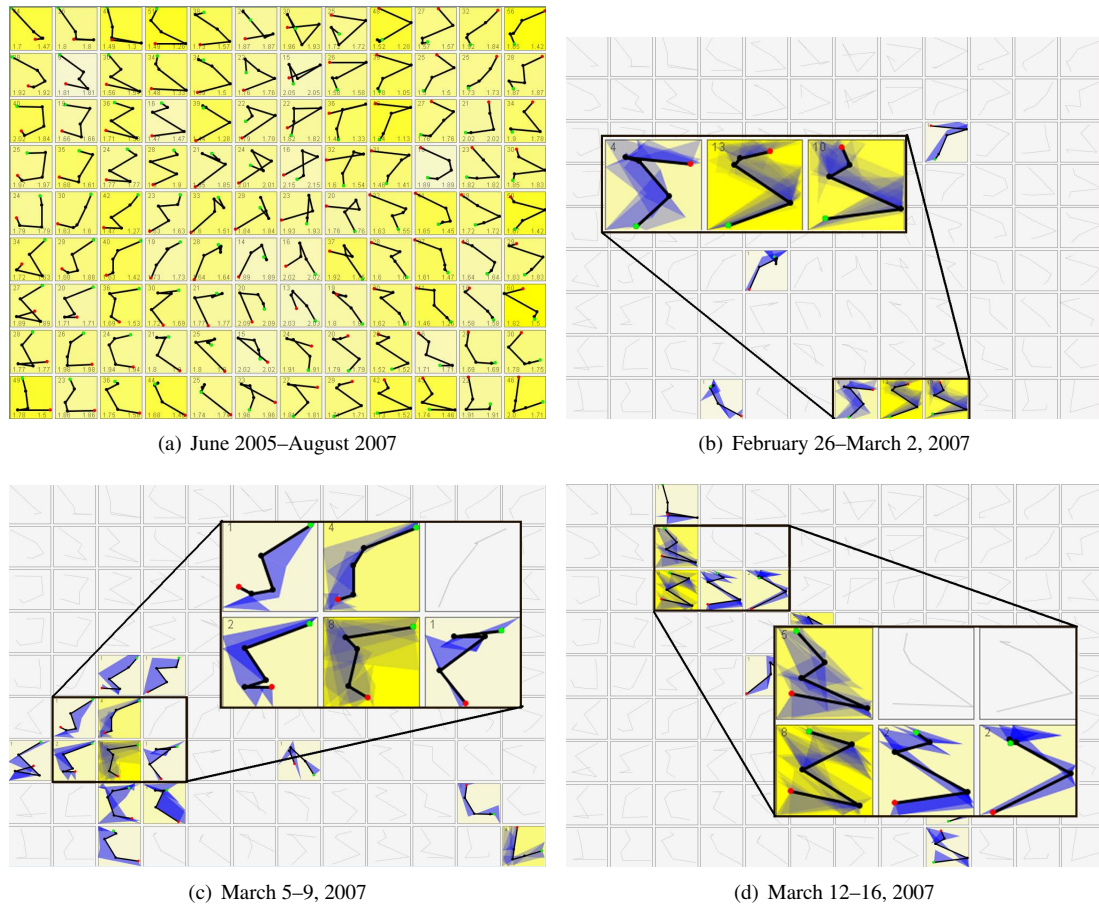


Figure 4.69.: Market movements in the selected time periods. Figure (a) shows an unsupervised clustering of weekly risk-return charts for 30 German blue chip stocks, as observed between 2005 and 2007. Figures (b) to (d) show a highlighted projection of the map to the chart patterns observed during three consecutive weeks, during a transitory downturn phase of the market (c,d; the most frequent patterns are zoomed in), preceded by a short upturn phase (b).

5. Conclusions and Future Challenges

This thesis discussed Visual Analytics methods for large data sets while concentrating on two selected data types: weighted directed graphs and two dimensional time dependent data. The area of application for the developed techniques was chosen to be finance and economics. Before turning to the more detailed conclusions and outlook for visual analysis of the two data types separately in Sections 5.2 and 5.3, it seems appropriate to first discuss the general results and foreseeable future challenges of our work.

5.1. General Remarks

5.1.1. Conclusions

As a background to our work, we discussed the methodology of Visual Analytics. In this respect, we compared the Visual Analytics research field with similar disciplines confirming their strong overlap. Visual Analytics is generally seen as an interdisciplinary field mainly combining methods from information visualization, interaction, data processing (especially data mining) and reasoning. Therefore, we methodologically examined three main interlocked disciplines in Visual Analytics: information visualization, interaction and data processing.

When concentrating on information visualization, we presented a new definition of data type space enhancing previous definitions with more data dimensions (time and uncertainty as special dimensions). This extension allows for definition also of complex data types such as time-dependent graphs and two-dimensional time-dependent data. Based on this new definition, a new categorization of information visualization techniques for Visual Analytics was presented. This extensive categorization (and implicitly survey) of techniques serves as a baseline for the proper choice of the techniques used in Visual Analytics systems. This survey has identified fields of potential future visualization research. Especially, the number and diversity of available techniques for time-dependent and compound graphs (compound graphs have both generic and hierarchic relations between nodes) is relatively small.

Additionally, a novel taxonomy of interaction techniques with focus on Visual Analytics was presented. While previous taxonomies concentrated on interaction in individual research areas, a unification of the taxonomies for Visual Analytics was needed. This unification focuses on two aspects: the interrelationship between the fields and unification of terminology in classification. In visual analytic tools, for example, owing to the integration of the visualization and data processing, the interaction in visual environment can automatically trigger start of a data processing algorithm whose results are directly visualized. In this case the user interaction in visualization also interacts with data processing, which needs to be considered. This taxonomy is useful for development of analysis tracking features within visual analytics systems.

In the main part of the thesis, we developed new methods for the visual analysis of two types of abstract data: weighted directed graphs and two-dimensional time-dependent data. Our general approach for the visual analysis of the two data types is based on the effective interactive visualization of the data combined with an appropriate algorithmic data analysis and tracking of the analysis process. Interactive visualization is suitable mainly for exploration of the data space. However with rising data set sizes, sole data exploration seems insufficient. Therefore, in order to gain insights into the data, the combination of selected algorithmic methods with interac-

tive exploration of the results is needed. In this respect, we follow Keim’s Visual Analytics Mantra [KKMT06]: “Analyse First – Show the Important – Zoom, Filter and Analyse Further – Details on Demand”. By flexible combination of the presented methods on demand a wide variety of possible user tasks can be covered. For better reproducibility of the results, we also included in some cases a tracking of the analytic process.

Although our general approach is similar for both types of the data, each data type requires specific data processing and visualization techniques. These were discussed and implemented in the thesis. For instance, we apply the well known self-organizing maps algorithm in order to reduce the data space into prominent data samples both for graphs and two-dimensional time dependent data. We follow similar processes of feature extraction, feature selection, SOM initialization and learning, visualization of clustering results and clustering quality in SOM grid. In this process, the specific features used and visualization techniques employed vary owing to the different data types.

The methods developed in our work can be used in a wide variety of applications. We focused on the financial and economic domains. In particular, the presented methods were applied to the analysis of shareholding networks and of time-dependent risk-return indicators of financial stocks. We thereby showed the suitability of the new methods for real world applications. The tools can support, for example, making of investment decisions, assessment of company value, financial market supervisory tasks and analysis of stock market developments. Our methods are, however, also applicable to other use cases in finance and economics such as cash flow analysis, supply chain management, monetary analysis, as well as to other areas including biology, transportation, or the wider social sciences.

5.1.2. Future Challenges

In this thesis, further steps towards the methodological basis of Visual Analytics and the adjacent fields together with practical approaches to the visual analysis of two types of data (weighted directed graphs and two-dimensional time-dependent data) were made. However, there is still a broad research potential open on both sides.

On the methodological side, it seems that the introduction of a comprehensive Visual Analytics model/process (analogous to information visualization pipeline [CMS99]), which would provide a more specific guidance for building Visual Analytics systems would be very useful. Currently, the Visual Analytics model of Keim [KMS*08] goes a long way in the envisaged direction of research, but still remains at a very abstract level. More details on steps in the development process suitable for clearly defined building blocks of Visual Analytics systems and their interrelation should be developed. In addition, with regard to the reasoning part of Visual Analytics systems, deeper understanding of the analytical process via an examination of the tracked parameters or psychological analysis could contribute to guidelines for building visual analytic systems with recommendation and/or user guidance components [TC06]. These systems could support the user in the selection of the appropriate analysis tools, propose further analytical steps based on the processes used for previous findings.

In the practical part of the thesis, we concentrated on visual analysis of two selected types of data. In the future, it would be interesting to examine the applicability of our approaches to additional data types and extend them appropriately. For example, as our overview of techniques in Section 2.3.3 shows, visual analysis of time-dependent graphs, compound graphs, or time-dependent multidimensional data seems a promising future direction of research. Along a different dimension, it would be interesting to develop integrated systems including data sets from various data sources and of multiple data types such as video, text, images 3D models or audio.¹. The tighter integration of such generalized digital documents offers possibilities for synergies via interlinkages of insights gained from the analysis of the individual data sets. In this sense, also the inclusion of the available

¹With regard to 3D documents, the research challenges have been recently stated in [HF07a].

expert knowledge and economic model sources as well as data semantics could be useful for gaining better analytical results. In this respect, the inclusion of knowledge representation in Visual Analytics systems becomes more necessary [Koh05]. Additionally, it has been shown [DE06] that data quality influences decision-making. Therefore inclusion of available data quality information in visualization and data processing is of relevance. However, this is still a generally recognized challenge in Visual Analytics area [LK07].

Turning to the evaluation of the thesis results: although we have documented the applicability of the methods on real world data sets and tasks including small qualitative user studies, broader and more detailed user studies of the presented Visual Analytics systems need to be conducted. This would be, however, an extensive issue which reaches beyond scope of this thesis – in particular when taking into consideration the still ongoing discussion in the Visual Analytics community on the appropriate methodology for the evaluation of Visual Analytics and information visualization systems. This methodological challenge is expressed in the statement by Plaisant et al. in the introduction to the special issue of Computer Graphics and Applications [PGS09] *“Assessing VA [Visual Analytics] technology’s effectiveness is challenging because VA tools combine several disparate components, both low and high level, integrated in complex interactive systems used by analysts, emergency responders, and others. ... Traditional evaluation metrics such as task completion time, number of errors, or recall and precision are insufficient to quantify the utility of VA tools, and new research is needed to improve our VA evaluation methodology.”*. The current research studies and community discussion fora on this issue include VisMaster Project www.vismaster.eu, Workshops “BEyond time and errors: novel evaluation methods for Information Visualization” at CHI 2008 <http://www.dis.uniroma1.it/~beliv08/> and “Metrics for the Evaluation of Visual Analytics” at VIS 2007 <http://www.cs.umd.edu/hcil/InfoVisworkshop/> or a special issue of Computer Graphics & Applications in June 2009).

Finally, the approaches for visual analysis of the data used in both parts of the thesis include techniques from data visualization, data processing and reasoning. These methods can be improved in various ways (see also Subsections 5.2.2 and 5.3.2). Specifically, in visual analysis of the two data types, interactive visual clustering has been used. The clustering methods are based on appropriate feature vector description of the input data and the usage of self-organizing maps. In the future, it would be of advantage to compare multiple clustering methods and expand SOM clustering with hierarchic means. The proposed input feature sets can be expanded with new features for gaining better quality of the results. Furthermore, sensitivity analysis of these features on the result can be performed and visual comparison of these results can be developed. The visual analysis of the clustering quality can be improved with new cell and data item specific methods. These issues are part of the recently started project “Visual Feature Space Analysis” in the context of DFG SPP “Scalable Visual Analytics” conducted at Interactive Graphics System Group at Technical University Darmstadt. Along a different line, when handling massive data sets, the usage of computing power of grid, cloud and GPU-based computing seem promising extensions for the efficiency of the Visual Analytics tools by decreasing the computational time.

5.2. Visual Analysis of Weighted Directed Graphs

For the visual analysis of weighted directed graphs, we have introduced a system which relies mainly on the combination of interactive visualization techniques with algorithmic data analysis on demand. The application of these techniques helps answering various analytical tasks. The developed techniques were applied on shareholding networks from Germany showing its applicability to real world tasks in financial industry.

5.2.1. Conclusions

Our approach for visual analysis of weighted directed graphs consist of three parts, which are integrated into a single system. Based on flexible combinations of these functions, the Visual Analytics system supports a wide range of analytic tasks.

Firstly, as a basis for the visual analysis, we presented an interactive visualization of weighted directed networks using state-of-the-art graph layouts supported by algorithmic graph analysis. This approach allows for the effective visual exploration of paths and connections between entities in the network. In the case of shareholding networks, visual identification of ultimate shareholders, integrated cash flows and control rights is addressed in particular. The highlighting of paths and connections allows for better orientation in the network which can be often difficult to analyze especially with growing number of nodes and edges. For very large networks, filtering of the important parts (the found connections) is of advantage although the context of the whole network may not be available then.

Secondly, we explored how the use of graph motifs can contribute to the visual analysis of large graphs. An extension of search and interactive visualization of predefined motifs with user-defined graph substructures was introduced. It offers the possibility to discover and search for both known and new/unknown structures in the data, which occur with high frequency or have specific function. Search for motifs can be used both for highlighting interesting parts in the whole network or as graph filter where only relevant parts of the large network are displayed.

For the analysis of graph changes, in particular for “what-if-analysis”, we extended current approaches by showing the implications of changes on local structures (motifs). In the state of the art displays only the current changes are highlighted. In the proposed way, also structural dependencies within the network can be discovered.

A new way of hierarchic graph aggregation based on motifs, which allows the analysis of graph structures and their relations at multiple abstraction levels was introduced. Motif-based graph aggregation allows for simplification of graphs while concentrating on inter-structural dependencies in the network. In this way, the relationships between functional substructures can be analyzed.

These functions are integrated with interactive visual exploration of the data. In the analysis of shareholder networks, it allows for assessment of structural features of company networks, examination of implications of company defaults or selling/buying of shares on company structure.

Thirdly, a novel approach for visual analysis of many graphs was proposed. In particular many weakly connected components of a graph were in focus of our work. This type of analysis has been largely disregarded in the graph visualization community. Our approach is based on an effective combination of adaptive graph clustering and rich visual-interactive facilities for data exploration. Interactive feature selection for flexible clustering with visual output, and assessment of clustering quality provide comprehensive visual cluster analysis for graph data. The reproducibility and comparability of cluster results is supported by storing analysis parameters and user annotations. These tools allow assessment of the structural compositions of a large set of graphs, for example, company networks in an economic system.

5.2.2. Future Challenges

The possible future directions of the work in the analysis of graphs relate to each of the three parts of our approach.

Firstly, our approach to graph visualization is currently layout independent using available graph layouts. As the understandability of the displayed data depends on the used layout algorithm, the visual exploration of the

graphs can be enhanced with new layout algorithms for large graphs. Development of graph layouts is thereby a specific research area not in the focus of the thesis.

The analysis of graphs is currently supported in our system by functions for detection of paths and connections between entities. The spectrum of offered functions could be extended with additional graph algorithms in order to broaden the scope of the analytical tasks covered such as degree of interest functions based on node centrality [vHP09].

Secondly, while our method for the visual analysis of graph motifs advances interactive motif- and change-based visual graph analysis, interesting problems that need to be solved exist on the sides of algorithmic motif analysis, aggregation, and visualization. In particular, in order to increase the efficiency of our approach, it is interesting to address the computational complexity of identification of motifs of arbitrary size and shape. This can be done by either development of new, more efficient search algorithms or by faster computation using, e.g., parallelization. Additionally, visualization of overlapping motifs and their aggregation are still open challenges. In particular, more motif-oriented visual representations including motif-based layouts could be developed. Furthermore, new methods for the visualization of multi-level aggregated graph could be included in our system. Currently, the aggregated nodes in the display replace the original nodes whereby the motif type used for aggregation is indicated by an icon. This approach could be extended with techniques for visualization of compound graphs such as [AMA08] or new techniques can be developed. In the analysis of graph changes (in particular in the what-if-analysis), our system currently shows changed motifs via highlighting/filtering, i.e., found motifs are contrasted against/selected from the overall graph in which they are embedded. When showing the motif changes, our approach should be adjusted on the general case, when several nodes/edges change at the same time and overlapping motif changes need to be visualized.

Thirdly, the visual analysis of sets of graphs employs interactive clustering and visual exploration of the results. Future challenges include clustering-based issues which have been already discussed in Section 5.1.2. In this respect however specific issues for graph analysis arise. For example, definition of features for further graph types such as labeled graphs can be included in the future thereby broadening the application scope of the approach also to, e.g., biologic networks. Additionally, the easy understanding of the clustering results can be increased by development of specialized layouts supporting comparison of graphs in the visualization.

5.3. Visual Analysis of Two-Dimensional Time-Dependent Data

For the visual analysis of two-dimensional time dependent data, we have presented a system combining interactive visualization and algorithmic data analysis. The usability of the system has been demonstrated on stock market data focusing on risk-return dynamics.

5.3.1. Conclusions

The approach for visual analysis of two-dimensional time dependent data presented in Chapter 4 includes three inter-related parts, which are summarized in the following.

Firstly, the interactive visualization of two dimensional time-dependent data uses a scatter plot framework to display relationships between two data dimensions. The system employs animation of glyphs and display of data trajectories to handle the time dimension of the data. The display includes, if appropriate for the task, additional dimensions of the data encoded by a suitable glyph design. Multiple interaction features are provided for the exploration of the data set and thereby enhance these analytical capabilities. Preliminary user experience confirmed the usability of the system in solving real-world tasks and, especially, the value added provided by

the animation and filtering features in the analysis process. Our results show that animation is more suitable for analysis of general or exceptional data dynamics and trajectories are more suitable for detailed examination of the movements. These results were also confirmed in [RFF*08]. Although our approach supports examination of movement of data items, for larger data sets reduction of the data space or focus on interesting parts of the data is needed. Specifically, as trajectories suffer from overplotting and animation from limited human perception capacities. Approaches for such reductions were presented in the following two parts of our approach.

With regard to the visual exploration of time-dependent data in a scatterplot framework using animation, the system settings need to take human perception issues into account. In order to examine the animation perception, we conducted a study on the ability to detect changes in the direction of the groups of entities. The results showed that the speed of animation has a major influence on the ability to draw conclusions from the displayed data. Therefore appropriate animation speeds should be used. Too slow or too fast animation results in longer reaction times and lower response accuracy. Since our results showed no significant effects of the angle of direction change, users are supposed to be able to interpret the data changes along major directions correctly. These results contribute to the guidelines for setting up animated 2D scatterplot-based visualization systems.

Secondly, for the visual analysis of two-dimensional time-dependent data with grouping information, we presented an approach which relied on a combination of a rich set of visual representations of both the input data and of extracted time-varying data features. In our work, we proposed new features and developed a systematization of the features for monitoring dynamics of a) entities within groups, b) individual groups and c) relationship of multiple groups. These features are able to identify a wide spectrum of data patterns in all three task groups. The results are used for selection of interesting data views for further exploration. In this way, the drawbacks of the first approach can be overcome. This tool can be used for analysis of intra- and inter-group dynamics such as of country developments in stock market data.

Thirdly, we presented a tool for pattern-based analysis of two-dimensional time-varying data. It uses trajectory fragmentation, feature extraction, SOM clustering and interactive visualization for finding patterns in the data. This system allows for identification and exploration of typical and extraordinary patterns in the trajectory data. In addition, we developed a visual-interactive framework for guiding the otherwise unsupervised Self-Organizing Map algorithm by a user. The framework enables the user to visually monitor the clustering process and control the algorithm at an arbitrary level of detail. The monitoring and interactive initialization support better understanding and interpretability of the results. These tools can be used in analysis of patterns of trajectories in general and stock market movements in particular.

5.3.2. Future work

The possible future directions of the work on visual analysis of two-dimensional time dependent data are connected to each of the three pillars of our approach and also to the overall approach.

Firstly, the perception study on the discrimination of directional changes in animated scatterplots can be extended by conducting experiments employing a greater range of velocities to find whether the 'fast'-velocity effect could be amplified and, conversely, at which velocity values the effect levels off. Furthermore, it would be interesting to examine smaller intervals of velocity values for finding out the pattern of reaction times and response accuracy with regard to the speed of motion. Moreover, experiments analyzing the effects of encoding of multiple visual variables (size, shape, color) in connection to animation could contribute to broader set of specific guidelines for the design of animated visualizations. In this respect, a recent publication [LvW09] has studied the effect of symbol mapping (glyph design) on the human perception for a static data presentation. It would be interesting to extend these experiments with animation and further settings of data mappings.

Secondly, our approach to the visual analysis of groups of two-dimensional time dependent data combines extraction of data features with interactive visualization. Interesting future work includes further broadening of the system with more analytic functions and improved visual representations for the different group features. The visual representation could be enhanced by combining statistic data properties with visual data abstractions (e.g., specific data symbols). The focus of analysis and the related set of features can be extended to cover also changing (time-varying) grouping of entities. It seems that the development of new and combination of the used features may bring additional information on the data behaviors, and analysis of the interplay of these indicators may lead to new interesting insights into the data. Additionally, it would be interesting to use time series analysis techniques for analysis of the extracted time varying features and thereby to automatically indicate interesting time intervals in the displayed data.

Thirdly, in the analysis of the patterns in data trajectories, we used clustering combined with visual exploration of the results. In this respect, the general possible improvements mentioned in Section 5.1.2 also refer to this approach. Specifically, with regard to visualization of clustering results of time dependent data, one of the views shows sequences of identified patterns over time. In this view, it would be interesting to include more advanced analysis algorithms from time series and sequence analysis for finding longer recurring sequences of patterns and visually explore them. These issues are part of the recently started project “Visual Feature Space Analysis” in the context of DFG SPP “Scalable Visual Analytics” conducted at Interactive Graphics System Group of Technical University Darmstadt. The first recent results are currently published in [BMvLS09]. Please note that future extensions to the clustering and visual analysis presented in Section 5.1.2 is also part of the same project. In the application, we used individual pattern normalization to $[0, 1]^2$, other normalizations (addressing other types of invariances) would in the future allow for tackling more domain specific tasks.

When looking at a general level, we presented a broad variety of methods for analyzing time dependent data, however these methods are focused on the interpretation of historical developments. In order to broaden the scope of the analytical tasks covered, the focus on methods for prediction of future development seems a necessary further extension.

A. Publications and Talks

The thesis is partially based on the following publications and talks:

A.1. Publications

1. Tekušová, Tatiana; Kohlhammer, Jörn: Applying Animation to the Visual Analysis of Financial Time-Dependent Data, In Proceedings of IEEE Computer Society: 11th International Conference on Information Visualisation, pp. 101–108, 2007
2. Schreck, Tobias; Tekušová, Tatiana; Kohlhammer, Jörn; Fellner, Dieter: Trajectory-Based Visual Analysis of Large Financial Time Series Data, ACM SIGKDD Explorations, Special Issue on Visual Analytics, December 2007, Volume 9, Issue 2, pp. 30–37 2007
3. Tekušová, Tatiana; Voss, Viktor: Semantic Search and Visualization of Time-Series Data, In Proceedings of I-KNOW '08, the 8th International Conference on Knowledge Management and I-MEDIA '08, International Conference on New Media Technology. pp. 332–340, 2008
4. Kalbe, Thomas; Tekušová, Tatiana; Schreck, Tobias and Zeilfelder, Frank: GPU-Accelerated 2D Point Cloud Visualization using Smooth Splines for Visual Analytics Applications, In Proceedings of Spring Conference on Computer Graphics, pp. 111–125, 2008
5. Tekušová, Tatiana; Schreck, Tobias: Visualizing Time-Dependent Data in Multivariate Hierarchic Plots - Design and Evaluation of an Economic Application, In Proceedings of IEEE Computer Society 12th International Conference on Information Visualisation, pp. 143–150, 2008
6. Tekušová, Tatiana; Knuth, Martin; Schreck, Tobias; Kohlhammer, Jörn: Data Quality Visualization for Multivariate Hierarchic Data, IEEE Information Visualization Conference, Poster Paper, pp. 108–109, 2008.
7. Tekušová, Tatiana; Skwarek, Slawomir; Kohlhammer, Jörn; Paramei, Galina: Perception of Direction Changes in Animated Data Visualization, In Proceedings of Symposium on Applied Perception in Graphics and Visualization, Poster Session, pp. 205–205, 2008
8. Schreck, Tobias; Bernard, Jürgen; Tekušová, Tatiana; Kohlhammer, Jörn: Visual cluster analysis in trajectory data using editable Kohonen Maps, In. Proceeding of IEEE Symposium on Visual Analytics Science and Technology, pp. 3–10, 2008
9. Tekušová, Tatiana; Kohlhammer, Jörn: Visual Analysis and Exploration of Complex Corporate Shareholder Networks, IS&T/SPIE Conference on Visualization and Data Analysis, Volume 6809, pp. 68090F, 2008
10. von Landesberger, Tatiana; Voss, Viktor and Kohlhammer, Jörn: Semantic Search and Visualization of Time-Series Data, Springer Series, Volume 221, p. 205–216, 2009
11. Schreck, Tobias; Bernard, Jürgen; von Landesberger, Tatiana; Kohlhammer, Jörn: Visual Cluster Analysis in Trajectory Data Using Editable Kohonen Maps, Information Visualization, Volume 8, pp. 14–29, 2008

12. von Landesberger, Tatiana; Görner, Melanie; Schreck, Tobias: Visual Analysis of Graphs with Multiple Connected Components, In Proceedings of IEEE Symposium on Visual Analytics Science and Technology, pp. 155–162, 2009
13. von Landesberger, Tatiana; Rehner, Robert; Görner, Melanie and Schreck, Tobias: A System for Interactive Visual Analysis of Large Graphs Using Motifs in Graph Editing and Aggregation, In Proceeding of Vision Modeling Visualization Workshop, pp. 331–339, 2009
14. von Landesberger, Tatiana; Bremm, Sebastian; Rezaei, Peyman and Schreck, Tobias: Visual Analytics of Time Dependent 2D Point Clouds, In Proceedings of Computer Graphics International, pp. 97–101, 2009
15. Bremm, Sebastian; Maier, Sebastian; von Landesberger, Tatiana and Schreck, Tobias: Ein flexibles System für die explorative visuelle Sequenzanalyse, Special Issue on Visual Analytics Datenbank Spektrum, pp. 8–16, December 2009
16. Bernard, Jürgen; von Landesberger, Tatiana; Bremm, Sebastian and Schreck, Tobias: Micro-Macro Views for Visual Trajectory Cluster Analysis. (Poster paper), Eurographics/IEEE Symposium on Visualization, 2009
17. Schreck, Tobias; von Landesberger, Tatiana and Bremm, Sebastian: Techniques for Precision-Based Visual Analysis of Projected Data, In Proceedings of IS&T/SPIE Electronic Imaging: Visualization and Data Analysis, Volume 7530, pp. 75300E-1–75300E-12, 2010, Best paper award
18. von Landesberger, Tatiana; Kuijper, Arjan; Schreck, Tobias; Kohlhammer, Jörn; van Wijk, Jarke J.; Fekete, Jean-Daniel and Fellner, Dieter W.: Visual Analysis of Large Graphs, Proceedings of Annual Conference of the European Association for Computer Graphics, pp. 113–136, 2010
19. Andrienko, Gennady; Andrienko, Natalia; Bremm, Sebastian; Schreck, Tobias; von Landesberger, Tatiana; Bak, Peter and Keim, Daniel A.: Space-in-Time and Time-in-Space Self-Organizing Maps for Exploring Spatiotemporal Patterns, Eurographics/IEEE Symposium on Visualization, pp. 913–922, 2010
20. von Landesberger, Tatiana; Bremm, Sebastian; Bernard, Jürgen; Schreck, Tobias: Smart Query Definition for Content-Based Search in Large Sets of Graphs, International Symposium on Visual Analytics Science and Technology, pp. 7–12, 2010

A.2. Talks

1. Tatiana Tekušová: Visual Analytics Tools for Analysis of Financial Data, ISBIS Conference, Prague, 2008
2. Tatiana von Landesberger: Visual Analytics Tools for Analysis of Financial Data, Invited Talk, Economic Risk Seminar, Humboldt University, Berlin, 2009

B. Curriculum Vitae

Personal Data

Name	Tatiana Landesberger von Antburg, maiden name: Tekušová
Birth date & place	08.06.1979 in Bratislava, Slovakia
Family status	Married
Nationality	Slovak

Education

2003	Graduation in Economic and Financial Mathematics at Comenius University, Bratislava, Slovakia
1997 – 2003	Study at Comenius University, Bratislava, Slovakia, Faculty of Mathematics, Physics and Informatics
2000 – 2001	Study at University of Regensburg, Germany, major: economics
July 2000	International Summer University, Vienna, Austria, Subjects: risk management and financial services, international comparative organization theory

Work Experience

04/2008 –	Researcher, Interactive Graphics Systems Group, Technische Universität Darmstadt, Germany, Focus: Visual Analytics for Financial Sector
04/2006 –	Researcher, Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany, Focus: Visual Analytics for Financial Sector
04/2004 – 03/2006	Research Analyst, European Central Bank, Frankfurt am Main, Germany, Monetary Policy Stance Division
11/2003 – 04/2004	Intern, European Central Bank, Frankfurt am Main, Germany, Monetary, Financial Institutions and Markets Statistics Division

Bibliography

- [AA07] ANDRIENKO N., ANDRIENKO G.: Designing visual analytics methods for massive collections of movement data. *Cartographica* 42, 2 (2007), 117–138. [19](#), [137](#), [145](#)
- [AA08] ANDRIENKO G., ANDRIENKO N.: Spatio-temporal aggregation for visual analysis of movements. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2008), pp. 51–58. [143](#), [145](#)
- [AAG00] ANDRIENKO N., ANDRIENKO G., GATalsky P.: Supporting visual exploration of object movement. In *Proceedings of the working conference on Advanced visual interfaces* (2000), pp. 217–220. [143](#)
- [AAK*08] ANDRIENKO G., ANDRIENKO N., KOPANAKIS I., LIGTENBERG A., WROBEL S.: *Mobility, Data Mining and Privacy*. Springer Berlin Heidelberg, 2008, ch. Visual Analytics Methods for Movement Data, pp. 375–410. [2](#), [144](#), [163](#), [164](#)
- [AAM07] ARCHAMBAULT D., AUBER D., MUNZNER T.: Topolayout: Multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 305–317. [29](#), [52](#), [63](#)
- [AAPS08] ANDRIENKO G., ANDRIENKO N., PELEKIS I., SPACCAPIETRA S.: *Mobility, Data Mining and Privacy*. Springer Berlin Heidelberg, 2008, ch. Basic Concepts of Movement Data, pp. 15–38. [140](#), [141](#), [163](#), [164](#), [172](#)
- [AAR*09] ANDRIENKO G., ANDRIENKO N., RINZIVILLO S., NANNI M., PEDRESCHI D., GIANNOTTI F.: Interactive visual clustering of large collections of trajectories. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 3–10. [137](#)
- [AAW07] ANDRIENKO G., ANDRIENKO N., WROBEL S.: Visual analytics tools for analysis of movement data. *SIGKDD Explorations* 9, 2 (December 2007), 38–46. [137](#)
- [ABM*07] AIGNER W., BERTONE A., MIKSCH S., TOMINSKI C., SCHUMANN H.: Towards a conceptual framework for visual analytics of time and time-oriented data. In *Proceedings of the 39th conference on Winter simulation* (2007), pp. 721–729. [25](#), [26](#)
- [ACJM03] AUBER D., CHIRICOTA Y., JOURDAN F., MELANCON G.: Multiscale visualization of small world networks. In *Proceedings of IEEE Symposium on Information Visualization* (2003), pp. 75–81. [50](#)
- [ADWM04] ADAI A. T., DATE S. V., WIELAND S., MARCOTTE E. M.: LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. *Journal of Molecular Biology* 340, 1 (June 2004), 179–190. [52](#)
- [AEF*95] AKKIRAJU N., EDELSBRUNNER H., FACELLO M., FU P., MUCKE E. P., VARELA C.: Alpha shapes: definition and software. In *Proceedings of International Computational Geometry Software Workshop* (1995). [167](#)
- [AF07] APPERT C., FEKETE J.-D.: Naviguer dans des grands arbres avec controltree. In *IHM '07: Proceedings of the 19th International Conference of the Association Francophone*

- d'Interaction Homme-Machine* (New York, NY, USA, 2007), ACM, pp. 139–142. 53
- [AH98] ANDREWS K., HEIDEGGER H.: Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proceedings of IEEE Symposium on Information Visualization* (1998). 29
- [AK07] ANDREWS K., KASANICKA J.: A comparative study of four hierarchy browsers using the hierarchical visualisation testing environment (HVTE). In *Proceedings of International Conference Information Visualization* (2007), pp. 81–86. 28
- [AKK96] ANKERST M., KEIM D. A., KRIEGEL H.-P.: Circle segments: A technique for visually exploring large multidimensional data sets. In *Proceedings of Visualization, Hot topics Session* (1996). 28
- [Alt06a] ALTANA AG: Altana Hauptversammlung stimmt Dividendenerhöhung um 16% zu. http://www.altana.com/de/pressarchiv_2006.php?id=736, May 2006. accessed 01.08.2009. 202
- [Alt06b] ALTANA AG: Altana mit hervorragendem 1. Quartal 2006. http://www.altana.com/de/pressarchiv_2006.php?id=724, April 2006. accessed 01.08.2009. 202
- [AMA08] ARCHAMBAULT D., MUNZNER T., AUBER D.: Grouseflocks: Steerable exploration of graph hierarchy space. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (July/August 2008), 900–913. 30, 53, 54, 221
- [AMA09] ARCHAMBAULT D., MUNZNER T., AUBER D.: Tuggraph: Path-preserving hierarchies for browsing proximity and paths in graphs. In *Proceedings of IEEE Pacific Visualization Symposium* (April 2009), pp. 113–120. 30, 53, 54
- [AMM*08] AIGNER W., MIKSCH S., MÜLLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 47–60. 143
- [Ank01] ANKERST M.: *Visual Data Mining*. PhD thesis, Ludwig Maximilian Universität München, 2001. 12
- [Arc09] ARCHAMBAULT D.: Structural differences between two graphs through hierarchies. In *Proceedings of Graphics Interface* (2009), pp. 87–94. 55
- [ASKP03] ALON J., SCLAROFF S., KOLLIOS G., PAVLOVIC V.: Discovering clusters in motion time-series data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2003), pp. 375–381. 142
- [AvH04] ABELLO J., VAN HAM F.: Matrix zoom: A visual interface to semi-external graphs. In *Proceedings of IEEE Symposium on Information Visualization* (2004), pp. 183–190. 53
- [AWS05] ALFRED-BUEHLER C., WATSON B., SHAMA D. A.: Visualizing live text streams using motion and temporal pooling. *IEEE Computer Graphics and Applications* 25, 3 (2005), 52–59. 155, 157
- [AWW09] ANDREWS K., WOHLFAHRT M., WURZINGER G.: Visual graph comparison. In *Proceedings of International Conference on Information Visualisation* (2009), pp. 62–67. 55
- [Bab02] BABURIN D. E.: Some modifications of Sugiyama approach. In *Revised Papers from International Symposium on Graph Drawing* (2002), pp. 366–367. 52
- [Bar08] BARTHEL K. U.: Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics. In *Proceedings of International Workshop on Image Analysis for Multimedia Interactive Services* (2008), pp. 227–230. 42

-
- [BB03] BRATH R., BRODY A.: Finding the needle in the haystack: Using data visualization to spot patterns and anomalies in business data. *Information Management Magazine* 2003, 10 (October 2003). 19, 43
- [BBD08] BURCH M., BECK F., DIEHL S.: Timeline trees: visualizing sequences of transactions in information hierarchies. In *Proceedings of the working conference on Advanced visual interfaces* (2008), pp. 75–82. 31
- [BBG*09] BLAAS J., BOTHA C., GRUNDY E., JONES M., LARAMEE R., POST F.: Smooth graphs for visual exploration of higher-order state transitions. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 969–976. 29
- [BBPS*04] BARRAT A., BARTHELEMY M., PASTOR-SATORRAS R., VESPIGNANI A., PARISI G.: The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of USA* 101, 11 (2004), 3747–3752. 87
- [BCB08] BIER E., CARD S., BODNAR J.: Entity-based collaboration tools for intelligence analysis. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 99–106. 19
- [BCLC97] BRODBECK D., CHALMERS M., LUNZER A., COTTURE P.: Domesticating bead: adapting an information visualization system to a financial institution. In *Proceedings of IEEE Symposium on Information Visualization* (1997), pp. 73–80. 23, 43
- [BD08] BURCH M., DIEHL S.: Timeradartrees: Visualizing dynamic compound digraphs. *Computer Graphics Forum* 27, 3 (2008), 823–830. 31
- [BDJ05] BRODLIE K. W., DUKE D. J., JOY K. I.: Arctrees: Visualizing relations in hierarchical data. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization* (2005), pp. 53–60. 30
- [Bel58] BELLMAN R. E.: On a routing problem. *Quarterly of Applied Mathematics* 16, 1 (1958), 87–90. 51
- [Ber81] BERTIN J.: *Graphics and Graphic Information-Processing*. Walter de Gruyter & Co, 1981. 22
- [BG07] BANG-JENSEN J., GUTIN G.: *Digraphs Theory, Algorithms and Applications*. Springer Verlag, 2007. 51
- [BGW03] BRANDES U., GÄRTLER M., WAGNER D.: Experiments on graph clustering algorithms. *Lecture Notes in Computer Science* 2832 (2003), 568–579. 50
- [BK04] BULIUNG R. N., KANAROGLOU P. S.: An exploratory spatial data analysis (ESDA) toolkit for the analysis of activity/travel data. *Lecture Notes in Computer Science* 3044 (2004), 1016–1025. 145
- [BKH05] BENDIX F., KOSARA R., HAUSER H.: Parallel sets: Visual analysis of categorical data. In *Proceedings of IEEE Symposium on Information Visualization* (2005), pp. 133–140. 27
- [BL09] BERTINI E., LALANNE D.: Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery* (2009), pp. 12–20. 14, 15, 16, 18, 33, 35, 37
- [BM02] BERTRAND M., MULLAINATHAN S.: *Pyramids*. Tech. Rep. 02–32, MIT Department of Economics Working Paper, 2002. 102
- [BMvLS09] BREMM S., MAIER S., VON LANDESBERGER T., SCHRECK T.: Ein flexibles system für die explorative visuelle Sequenzanalyse. *Datenbank Spektrum*, 31 (December 2009), 8–16. 223
-

- [BMZ*06] BRENNAN S., MÜLLER K., ZELINSKY G., RAMAKRISHNAN I., WARREN D., KAUFMAN A.: Toward a multi-analyst, collaborative framework for visual analytics. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (November 2006), pp. 129–136. [19](#)
- [BN01] BARLOW T., NEVILLE P.: A comparison of 2-D visualizations of hierarchies. In *Proceedings of IEEE Symposium on Information Visualization* (2001), pp. 131–138. [23](#), [28](#)
- [Bon87] BONACICH P.: Power and centrality: A family of measures. *The American Journal of Sociology* 92, 5 (1987), 1170–1182. [87](#)
- [BS87] BALL K., SEKULER R.: Direction-specific improvement in motion discrimination. *Vision Research* 27, 6 (1987), 953–966. [157](#)
- [Bun08] BUNDESANSTALT FÜR FINANZDIENSTLEISTUNGS AUFSICHT: www.bafin.de, accessed on 1.9.2008. [104](#)
- [Bur] BUREAU VAN DIJK: Amadeus database. <https://amadeus.bvdep.com>. accessed on 1.9.2009. [99](#)
- [BZL*08] BARLOWE S., ZHANG T., LIU Y., YANG J., JACOBS D.: Multivariate visual explanation for high dimensional datasets. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 147–154. [19](#)
- [Cal07] CALDARELLI G.: *Scale-Free Networks: Complex Webs in Nature and Technology*. Oxford Finance, 4 2007. [50](#), [51](#), [71](#), [87](#)
- [CC07] COLLINS C., CARPENDALE S.: VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1192–1199. [55](#)
- [CCR03] CAPOCCI A., CALDARELLI G., RIOS P. D. L.: Quantitative description and modeling of real networks. *Physical Review E* 68, 4 (2003), 047101. [87](#)
- [CGK*07] CHANG R., GHONIEM M., KOSARA R., RIBARSKY W., YANG J., SUMA E., ZIEMKIEWICZ C., KERN D., SUDJANTO A.: WireVis: Visualization of categorical, time-varying data from financial transactions. *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2007), 155–162. [19](#), [23](#), [43](#), [44](#)
- [CGMS03] CHUDOVA D., GAFFNEY S., MJOLSNES E., SMYTH P.: Translation-invariant mixture models for curve clustering. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2003), pp. 79–88. [142](#)
- [CGS00] CADEZ I., GAFFNEY S., SMYTH P.: A general probabilistic framework for clustering individuals. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2000), pp. 140–149. [142](#)
- [CGS03] CHIRINKO R., GARRETSEN H., STERKEN E.: *Corporate control mechanisms, voting and cash flow rights, and the performance of Dutch firms*. Working Paper 200309, University of Groningen, Centre for Economic Research, 2003. [102](#)
- [Cha09] CHABOT C.: Demystifying visual analytics. *Computer Graphics and Applications, IEEE* 29, 2 (March–April 2009), 84–87. [13](#)
- [Che08] CHEN C.: An information-theoretic view of visual analytics. *IEEE Computer Graphics and Applications* 28, 1 (Jan.–Feb. 2008), 18–23. [19](#)
- [Chi00] CHI E. H.: A taxonomy of visualization techniques using the data state reference model. In *Proceedings of IEEE Symposium on Information Visualization* (2000), pp. 69–78. [22](#)

-
- [Cia04] CIARDI F. C.: sMAX: A multimodal toolkit for stock market data sonification. In *Proceedings of Meeting of the International Conference on Auditory Display* (2004). [22](#), [23](#)
- [CK03] CRAIG P., KENNEDY J.: Coordinated graph and scatter-plot views for the visual exploration of microarray time-series data. In *Proceedings of IEEE Symposium on Information Visualization* (2003), pp. 173–180. [143](#), [149](#)
- [CL03] CHEN K., LIU L.: A visual framework invites human into the clustering process. In *Proceedings of International Conference on Scientific and Statistical Database Management* (July 2003), pp. 97–106. [42](#), [63](#), [151](#)
- [CMS99] CARD S. C., MACKINLAY J., SHNEIDERMAN B.: *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, 1999. [16](#), [21](#), [22](#), [23](#), [34](#), [37](#), [53](#), [218](#)
- [CN04] CHEN L., NG R.: On the marriage of Lp-norms and edit distance. In *Proceedings of International conference on Very large data bases* (2004), pp. 792–803. [172](#)
- [COO05] CHEN L., ÖZSU T. M., ORIA V.: Robust and fast similarity search for moving object trajectories. In *Proceedings of ACM SIGMOD International Conference on Management of Data* (2005), pp. 491–502. [142](#), [172](#)
- [CPC09] COLLINS C., PENN G., CARPENDALE S.: Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov.-Dec. 2009), 1009–1016. [28](#), [144](#)
- [CS02] CHAPELLE A., SZAFARZ A.: *Ownership and Control: Dissecting the Pyramid*. Working paper WP-CEB 03/002, Centre Emile Bernheim, Brussel, 2002. [102](#)
- [CW01] CHATZINIKOS F., WRIGHT H.: Computational steering by direct image manipulation. In *Proceedings of the Vision Modeling and Visualization Conference* (2001), Aka GmbH, pp. 455–462. [12](#)
- [CXGH08] CHAN S.-M., XIAO L., GERTH J., HANRAHAN P.: Maintaining interactivity while exploring massive time series. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008). [19](#)
- [CYR09] CHEN Y., YANG J., RIBARSKY W.: Toward effective insight management in visual analytics systems. In *Proceedings of IEEE Pacific Visualization Symposium* (April 2009), pp. 49–56. [19](#)
- [CZQ*08] CUI W., ZHOU H., QU H., WONG P. C., LI X.: Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1277–1284. [29](#), [63](#), [137](#), [144](#)
- [Dam05] DAMODARAN A.: *Dealing with Cash, Cross Holdings and Other Non-Operating Assets: Approaches and Implications*. Working paper, Stern School of Business, September 2005. [98](#)
- [DBETT99] DI BATTISTA G., EADES P., TAMASSIA R., TOLLIS I. G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999. [29](#), [52](#)
- [DBPBS07] DAO H. T., BAZINET A., P. BERTHIER R., SHNEIDERMAN B.: *NASDAQ Velocity and Forces: An Interactive Visualization of Activity and Change*. Tech. Rep. HCIL-2007-29, Human Computer Interaction Lab, Institute for Advanced Computer Studies, University of Maryland, 2007. [23](#)
- [DCCW08] DÖRK M., CARPENDALE S., COLLINS C., WILLIAMSON C.: VisGets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (November-December 2008), 1205–1212. [23](#)
-

- [DE02] DWYER T., EADES P.: Visualising a fund manager flow graph with columns and worms. In *Proceedings of Sixth International Conference on Information Visualisation* (10–12 July 2002), pp. 147–152. [23](#)
- [DE06] DEITRICK S., EDSALL R.: The influence of uncertainty visualization on decision making: An empirical evaluation. *Progress in Spatial Data Handling* (2006), 719–738. [219](#)
- [Deu] DEUTSCHE BOERSE: DAX. <http://deutsche-boerse.com/>. accessed 1.12.2009. [193](#)
- [DGK01] DIEHL S., GOERG C., KERREN A.: Preserving the mental map using foresighted layout. In *Proceedings of Joint Eurographics, IEEE TCVG Symposium on Visualization* (2001), pp. 175–184. [31](#)
- [DHKS05] DAYAL U., HAO M., KEIM D., SCHRECK T.: Importance driven visualization layouts for large time-series data. In *Proceedings of IEEE Symposium on Information Visualization* (2005), Stasko J., Ward M., (Eds.), IEEE Computer Society. [31](#)
- [Die05] DIESTEL R.: *Graph Theory*. Springer-Verlag, Heidelberg, 2005. [3](#), [48](#), [49](#), [50](#), [87](#)
- [Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik I* (1959), 269–271. [51](#)
- [DK97] DAVIS T. J., KELLER C. P.: Modelling and visualizing multiple spatial uncertainties. *Computers & Geosciences* 23, 4 (May 1997), 397–408. [32](#)
- [DK05] DWYER T., KOREN Y.: DIG-COLA: Directed graph layout through constrained energy minimization. In *Proceedings of IEEE Symposium on Information Visualization* (2005), pp. 65–72. [52](#)
- [DMS*08] DWYER T., MARRIOTT K., SCHREIBER F., STUCKEY P., WOODWARD M., WYBROW M.: Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov.-Dec. 2008), 1293–1300. [52](#)
- [DMW09a] DWYER T., MARRIOTT K., WYBROW M.: Dunnart: A constraint-based network diagram authoring tool. *Graph Drawing 5417* (2009), 420–431. [52](#), [53](#)
- [DMW09b] DWYER T., MARRIOTT K., WYBROW M.: Topology preserving constrained graph layout. In *Revised Papers from International Symposium on Graph Drawing* (2009), pp. 230–241. [52](#)
- [DO88] DE BRUYN B., ORBAN G. A.: Human velocity and direction discrimination measured with random dot patterns. *Vision Research* 28, 12 (1988), 1323–1335. [157](#), [162](#)
- [Dog02] DOGRUSOZ U.: Two-dimensional packing algorithms for layout of disconnected graphs. *Information Sciences* 143, 1–4 (2002), 147–158. [52](#)
- [DPS02] DÍAZ J., PETIT J., SERNA M.: A survey of graph layout problems. *ACM Comput. Surv.* 34, 3 (2002), 313–356. [29](#), [52](#)
- [DS82] D. TYNAN P., SEKULER R.: Motion processing in peripheral vision: Reaction time and perceived velocity. *Vision Research* 22, 1 (1982), 61–68. [162](#)
- [EC01] EWING R. M., CHERRY J. M.: Visualization of expression clusters using Sammons non-linear mapping. *Bioinformatics* 17, 7 (2001), 658–659. [42](#)
- [EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1539–1148. [23](#), [27](#)
- [EDG*08] ELMQVIST N., DO T.-N., GOODELL H., HENRY N., FEKETE J.-D.: Zame: Interactive large-scale graph visualization. In *Proceedings of IEEE Pacific Visualization Symposium* (2008),

- pp. 215–222. 29, 50, 53
- [EKS83] EDELSBRUNNER H., KIRKPATRICK D. G., SEIDEL R.: On the shape of a set of points in the plane. *IEEE Transactions on Information Theory* 29 (1983), 551–559. 167
- [EM95] ESTEBAN M., MORALES D.: A summary of entropy statistics. *Kybernetika* 31, 4 (1995), 337–346. 184
- [EN75] EVERITT B. S., NICHOLLS P.: Visual techniques for representing multivariate data. *The Statistician* 24, 1 (1975), 37–49. 27, 28
- [EST07] ELMQVIST N., STASKO J., TSIGAS P.: Datameadow: A visual canvas for analysis of large-scale multivariate data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (30 2007–Nov. 1 2007), pp. 187–194. 19, 27
- [Eur07] EUROPEAN CORPORATE GOVERNANCE INSTITUTE: *Report on the Proportionality Principle in the European Union*. Tech. rep., European Corporate Governance Institute, May 2007. http://www.ecgi.org/osov/documents/final_report_en.pdf. 1, 98
- [FAS04] FAS RESEARCH: Shareholder network of Austrian companies. In *Ars Electronica 2004* (2004). 98
- [FB94] FURNAS G. W., BUJA A.: Projection views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics* 3 (1994), 323–385. 27
- [FCI05] FANEA E., CARPENDALE S., ISENBERG T.: An interactive 3D integration of parallel coordinates and star glyphs. In *Proceedings of IEEE Symposium on Information Visualization* (2005), p. 20. 30
- [FCL*05] FU T.-C., CHUNG K. F.-L., LAM C.-F., LUK R. W. P., MAN NG C.: Adaptive data delivery framework for financial time series visualization. In *Proceedings of International Conference on Mobile Business* (2005), pp. 267–273. 43
- [FDK02] FREIVALDS K., DOGRUSÖZ U., KIKUSTS P.: Disconnected graph layout and the polyomino packing approach. In *Revised Papers from the 9th Int. Symposium on Graph Drawing* (2002), pp. 378–391. 52, 63
- [FdOL03] FERREIRA DE OLIVEIRA M., LEVKOWITZ H.: From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on* 9, 3 (July–Sept. 2003), 378–394. 12, 15, 21
- [FFG*08] FIANNACA A., FATTA G., GAGLIO S., RIZZO R., URSO A.: Clustering quality and topology preservation in fast learning soms. In *Proceedings of International Conference on Artificial Neural Networks, Part I* (2008), pp. 583–592. 42
- [FHK*09] FUNG D. C. Y., HONG S.-H., KOSCHUTZKI D., SCHREIBER F., XU K.: Visual analysis of overlapping biological networks. In *Proceedings of the International Conference Information Visualisation* (2009), pp. 337–342. 55
- [FHRH08] FISHER D., HOFF A., ROBERTSON G., HURST M.: Narratives: A visualization to track narrative events as they develop. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 115–122. 19, 31
- [FLM95] FRICK A., LUDWIG A., MEHLDAU H.: A fast adaptive layout algorithm for undirected graphs. In *Proceedings of DIMACS International Workshop on Graph Drawing* (1995), pp. 388–403. 52
- [Flo62] FLOYD R. W.: Algorithm 97 (shortest path). *Communications of the ACM* 5 (1962), 345. 51

- [For56] FORD L. R.: *Network flow theory*. Tech. Rep. P923, The Rand Corporation, 1956. 51
- [FPsS96] FAYYAD U., PIATETSKY-SHAPIO G., SMYTH P.: From data mining to knowledge discovery in databases. *AI Magazine* 17 (1996), 37–54. 13, 15, 16, 40
- [FR91] FRUCHTERMAN T. M. J., REINGOLD E. M.: Graph drawing by force-directed placement. *Software – Practice & Experience* 21, 11 (1991), 1129–1164. 52, 67
- [Fra06] FRANKFURTER ALLGEMEINE ZEITUNG: Schering lehnt Übernahmeangebot von Merck ab. <http://www.faz.net/s/RubE3BF7B6B26F443E5990C9BA42301A0C9/Doc~E484BC97407B241ACAE910D23C3085ECB~ATpl~Ecommon~Scontent~Afor~Eprint.html>, March 2006. accessed 01.08.2009. 202
- [Fre79] FREEMAN L. C.: Centrality in social networks. *Social Networks* 1, 3 (1979), 215–239. 54
- [FRE80] FRENCH K. R.: Stock returns and the weekend effect. *Journal of Financial Economics* 8 (1980), 55–69. 210
- [FT04] FRISHMAN Y., TAL A.: Dynamic drawing of clustered graphs. In *Proceedings of the IEEE Symposium on Information Visualization* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 191–198. 31
- [FT07] FRISHMAN Y., TAL A.: Multi-level graph layout on the gpu. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1310–1319. 52
- [FT08] FRISHMAN Y., TAL A.: Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 727–740. 31
- [FWD*03] FEKETE J.-D., WANG D., DANG N., ARIS A., PLAISANT C.: Overlaying graph links on treemaps. In *Proceedings of IEEE Information Visualization Symposium Posters Compendium* (2003), pp. 82–83. 30
- [FWR99] FUA Y.-H., WARD M., RUNDENSTEINER E.: Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Conference on Visualization* (1999), pp. 43–50. 42, 43, 189
- [Gaf03] GAFFNEY S. J.: Curve clustering with random effects regression mixtures. In *Proceedings of International Workshop on Artificial Intelligence and Statistics* (2003). 142
- [Gaf04] GAFFNEY S. J.: *Probabilistic Curve-Aligned Clustering and Prediction with Regression Mixture Models*. Tech. rep., Ph.D. Dissertation, 2004. Laboratoire MAS, 2004. 142
- [Gap] GAPMINDER: Gapminder. <http://www.gapminder.org/>. accessed 1.12.2009. 30, 143
- [GB02] GÜNTHER S., BUNKE H.: Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters* 23, 4 (2002), 405–417. 55
- [GBD09] GREILICH M., BURCH M., DIEHL S.: Visualizing the evolution of compound digraphs with TimeArcTrees. *Computer Graphics Forum* 28, 3 (2009), 975–982. 31
- [GBH98] GROS B. L., BLAKE R., HIRIS E.: Anisotropies in visual motion perception: a fresh look. *Journal of the Optical Society of America* 15, 8 (1998), 2003–2011. 157
- [GBPD04] GÖRG C., BIRKE P., POHL M., DIEHL S.: Dynamic graph drawing of sequences of orthogonal and hierarchical graphs. In *Graph Drawing* (2004), pp. 228–238. 31
- [GDLP09] GIACOMO E., DIDIMO W., LIOTTA G., PALLADINO P.: Visual analysis of one-to-many matched graphs. In *Revised Papers from International Symposium on Graph Drawing* (2009), pp. 133–144. 55

-
- [GF01] GHONIE M., FEKETE J.-D.: Animating treemaps. In *Proceedings of Workshop on Treemap Implementations and Applications* (2001). 31
 - [GFC04] GHONIE M., FEKETE J.-D., CASTAGLIOLA P.: A comparison of the readability of graphs using node-link and matrix-based representations. In *Proceedings of IEEE Symposium on Information Visualization* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 17–24. 29, 46, 52
 - [GH09] GANSNER E. R., HU Y.: Efficient node overlap removal using a proximity stress model. *Graph Drawing 5417* (2009), 206–217. 29
 - [GK01] GAJER P., KOBOUROV S. G.: GRIP: Graph drawing with intelligent placement. In *Proceedings of International Symposium on Graph Drawing* (2001), pp. 222–228. 52, 63
 - [GK07] GROCHOW J., KELLIS M.: Network motif discovery using subgraph enumeration and symmetry-breaking. *Research in Computational Molecular Biology* (2007), 92–106. 51, 72, 73, 74, 76
 - [GKS07] GÖHLS DORF D., KAUFMANN M., SIEBENHALLER M.: Placing connected components of disconnected graphs. In *Proceedings of Asia-Pacific Symposium on Information Visualisation* (Februar 2007), pp. 101–108. 52, 63
 - [Gof99] GOFFE B.: *Visualizing Multi-Dimensional Functions in Economics*. Computing in Economics and Finance 1334, Society for Computational Economics, March 1999. 43
 - [Gon96] GONZALEZ C.: Does animation in user interfaces improve decision making? In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems: Common Ground* (1996), pp. 27–34. 155
 - [GS66] GREEN D. M., SWETS J. A.: *Signal Detection Theory*. New York: Wiley, 1966. 160
 - [GS99] GAFFNEY S., SMYTH P.: Trajectory clustering with mixtures of regression models. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (1999), pp. 63–72. 142
 - [GS05] GRIETHE H., SCHUMANN H.: Visualizing uncertainty for improved decision making. In *Proceedings of International Conference on Perspectives in Business Informatics Research* (2005). 32
 - [GS06] GRIETHE H., SCHUMANN H.: The visualization of uncertain data: Methods and problems. In *Proceedings of Simulation and Visualization* (2006), pp. 143–156. 32
 - [GZ08] GOTZ D., ZHOU M.: Characterizing users’ visual analytic activity for insight provenance. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 123–130. 15, 16, 19, 33, 35, 37
 - [HB05] HEER J., BOYD D.: Vizster: visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on* (2005), pp. 32–39. 54
 - [HDK08] HAO M. C., DAYAL U., KEIM D. A.: Visual analytics techniques for large multi-attribute time series data. In *Proceedings of the SPIE Visualization and Data Analysis 2008* (jan 2008), vol. 6809, p. 680908. 31
 - [HDKS07] HAO M., DAYAL U., KEIM D., SCHRECK T.: A visual analysis of multi-attribute data using pixel matrix displays. In *Proceedings of IS&T/SPIE Conference on Visualization and Data Analysis* (2007), no. 649505. 28
 - [HE03] HOPF M., ERTL T.: Hierarchical splatting of scattered data. In *Proceedings of IEEE Visualization* (2003), pp. 433–440. 144
-

- [HEy] HEYEWALL SYSTEM, FRAUNHOFER IGD DARMSTADT, GERMANY: <http://www.heyewall.de/>. accessed 01.12.2009. 184
- [HF06] HENRY N., FEKETE J.-D.: MatrixExplorer: a dual-representation system to explore social networks. *Visualization and Computer Graphics, IEEE Transactions on* 12, 5 (Sept.-Oct. 2006), 677–684. 29, 53
- [HF07a] HAVEMANN S., FELLNER D. W.: Seven research challenges of generalized 3D documents. *IEEE Computer Graphics and Applications* 27, 3 (2007), 70–76. 218
- [HF07b] HENRY N., FEKETE J.-D.: Matlink: Enhanced matrix visualization for analyzing social networks. In *Proceedings of the International Conference Interact* (2007). 29, 30, 54
- [HFM07] HENRY N., FEKETE J.-D., MCGUFFIN M.: NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov.-Dec. 2007), 1302–1309. 29, 53
- [HH05] HUBER D. E., HEALEY C. G.: Visualizing data with motion. In *Proceedings of IEEE Visualization Conference* (2005), pp. 527–534. 157, 162
- [HHW05] HUANG Y.-P., HSU C.-C., WANG S.-H.: Visualizing efficiency and reference relations in data envelopment analysis with an application to the branches of a German bank. *Journal of Productivity Analysis* 23, 2 (2005), 203–221. 43
- [HHWN02] HAVRE S., HETZLER E., WHITNEY P., NOWELL L.: Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 9–20. 31
- [HJ05] HACHUL S., JÜNGE M.: Drawing large graphs with a potential-field-based multilevel algorithm. *Lecture notes in Computer Science* 4372 (2005), 285–295. 52
- [HJ07] HACHUL S., JÜNGER M.: Large-graph layout algorithms at work: An experimental study. *Journal of Graph Algorithms and Applications* 11, 2 (2007), 234–369. 29, 52
- [HK95] HAWAWINI G., KEIM D. B.: On the predictability of common stock returns: World-wide evidence. *Handbooks in Operations Research and Management Science* 9 (1995), 497–544. 210
- [HK98] HERRMANN A., KEIM D.: The Gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data. In *Proceedings of IEEE Visualization (VIS 1998)* (1998), pp. 181–188. 149
- [HK02] HAREL D., KOREN Y.: Graph drawing by high-dimensional embedding. In *Revised Papers from International Symposium on Graph Drawing* (2002), pp. 207–219. 52
- [HK03] HÖPNER M., KREMPEL L.: *The Politics of the German Company Network*. MPIfG Working Paper 03/9, Max Planck Institute for the Study of Societies, September 2003. 98
- [HK06] HAN J., KAMBER M.: *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kauffman, 2006. 16, 39, 40, 41
- [HKLK97] HONKELA T., KASKI S., LAGUS K., KOHONEN T.: WEBSOM—self-organizing maps of document collections. In *Proceedings of Workshop on Self-Organizing Maps* (1997), pp. 310–315. 42
- [HM95] HACKSTADT S. T., MALONY A. D.: Visualizing parallel programs and performance. *IEEE Computer Graphics and Applications* 15, 4 (1995), 12–14. 27

- [HM98] HOHNSBEIN J., MATEEFF S.: The time it takes to detect changes in the speed and direction of visual motion. *Vision Research* 38, 17 (1998), 2569–2573. 157, 162
- [HMM00] HERMAN I., MELANCON G., MARSHALL M. S.: Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics* 6, 1 (2000), 24–43. 28, 48, 52
- [HMS*05] HUANG W., MURRAY C. X. S., SONG L., WU Y. X., ZHENG L.: Visualisation and analysis of network motifs. In *Proceedings of International Conference on Information Visualisation* (July 2005), pp. 697–702. 55
- [Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 741–748. 29, 30, 182
- [Hop] HOPPENSTEDT: Hoppenstedt Konzernstrukturen. www.hoppenstedt-konzernstrukturen.de. accessed on 1.9.2009. 99, 102
- [HS04] HOCHHEISER H., SHNEIDERMAN B.: Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization* 3, 1 (2004), 1–18. 30
- [HTF09] HASTIE T., TIBSHIRANO R., FRIEDMAN J.: *The elements of Statistical Learning*, 2nd ed. Springer, 2009. 41
- [HvW08] HOLTEN D., VAN WIJK J. J.: Visual comparison of hierarchically organized data. *Comput. Graph. Forum* 27, 3 (2008), 759–766. 55
- [HvW09] HOLTEN D., VAN WIJK J. J.: A user study on visualizing directed edges in graphs. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 2299–2308. 29
- [HZG05] HOLLEIS P., ZIMMERMANN T., GMACH D.: Drawing graphs within graphs. *Journal of Graph Algorithms and Applications* 9, 1 (October 2005), 7–18. 55
- [IAG*09] IMAMICHI T., ARAHORI Y., GIM J., HONG S.-H., NAGAMUCHI H.: Removing node overlaps using multi-sphere scheme. In *Graph Drawing* (2009), pp. 296–301. 29
- [Ise07] ISENBERG P.: Information visualization in co-located collaborative environments. In *Proceedings of the Grace Hopper Celebration of Women in Computing, PhD Forum* (2007). 22
- [IWSK07] IVANOV Y., WREN C., SOROKIN A., KAUR I.: Visualizing the history of living spaces. *Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1153–1160. 19, 137
- [JA07] JAMIESON R., ALEXANDROV V.: A data forest: Multi-dimensional visualization. *Proceedings of International Conference on Information Visualisation* (2007), 293–300. 22
- [JHGH08] JIA Y., HOBEROCK J., GARLAND M., HART J.: On the visualization of social and other scale-free networks. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1285–1292. 50
- [JS03] JOHNSON C. R., SANDERSON A. R.: A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications* 23, 5 (Sept. 2003), 6–10. 32
- [JT92] JUNGMEISTER W., TURO D.: *Adapting treemaps to stock portfolio visualization*. UMCP-CSD CS-TR-2996 1996, University of Maryland, November 1992. 23
- [JTS08] JOHN M., TOMINSKI C., SCHUMANN H.: Visual and analytical extensions for the table lens. In *Proceedings of the SPIE Visualization and Data Analysis 2008* (jan 2008), vol. 6809, p. 680907. 28

- [KAF*08] KEIM D., ANDRIENKO G., FEKETE J.-D., GÖRG C., KOHLHAMMER J., MELANCON G.: *Information Visualization*, vol. 4950 of *Lecture Notes in Computer Science*. Springer, 2008, ch. Visual Analytics: Definition, Process, and Challenges, pp. 154–175. [12](#), [13](#), [15](#), [16](#), [17](#)
- [KAK95] KEIM D. A., ANKERST M., KRIEGEL H.-P.: Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of IEEE Conference on Visualization* (1995), p. 279. [28](#)
- [Kan01] KANDOGAN E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2001), pp. 107–116. [42](#)
- [Kas97] KASKI S.: *Data Exploration Using Self-Organizing Maps*. PhD thesis, Acta Polytechnica Scandinavica, Mathematics, Finland, 1997. [42](#)
- [KCH02] KOREN Y., CARMEL L., HAREL D.: ACE: A fast multiscale eigenvectors computation for drawing huge graphs. In *Proceedings of the IEEE Symposium on Information Visualization* (2002), p. 137. [52](#)
- [Kee06] KEEL P.: Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (November 2006), pp. 137–144. [19](#)
- [Kei02] KEIM D. A.: Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 1–8. [21](#), [22](#), [23](#)
- [Keo02] KEOGH E.: Exact indexing of dynamic time warping. In *Proceedings of International Conference on Very Large Data Bases* (2002), pp. 406–417. [172](#)
- [KG06] KUMAR G., GARLAND M.: Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sept.-Oct. 2006), 805–812. [31](#), [43](#)
- [KGS*08] KANG H., GETOOR L., SHNEIDERMAN B., BILGIC M., LICAMELE L.: Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Transactions on Visualization and Computer Graphics* (Mar 2008). [19](#)
- [KHD02] KEIM D. A., HAO M., DAYAL U.: Hierarchical pixel bar charts. *IEEE Transactions On Visualisation And Computer Graphics* 8, 3 (July 2002), 255–269. [29](#)
- [KHG03] KOSARA R., HAUSER H., GRESH D. L.: An interaction view on information visualization. In *State-of-the-Art Proceedings of EUROGRAPHICS* (2003), pp. 123–137. [53](#)
- [KHKL96] KOHONEN T., HYNINEN J., KANGAS J., LAAKSONEN J.: *SOM_PAK: The Self-Organizing Map Program Package*. Tech. Rep. A31, Helsinki University of Technology, 1996. [178](#), [211](#)
- [KHL*01] KEIM D., HAO M., LADISCH J., HSU M., DAYAL U.: Pixel bar charts: a new technique for visualizing large multi-attribute data sets without aggregation. *Information Visualization 2001* (2001), 113–120. [27](#), [28](#), [31](#)
- [KK89] KAMADA T., KAWAI S.: An algorithm for drawing general undirected graphs. *Information Processing Letters* 31, 1 (1989), 7–15. [52](#), [67](#)
- [KKMT06] KEIM D., KOHLHAMMER J., MAY T., THOMAS J.: Event summary of the workshop on visual analytics: June 4, 2005, darmstadt germany jointly organized by university of konstanz and fraunhofer igd. *Computers & Graphics* 30, 2 (2006), 284–286. [63](#), [151](#), [218](#)
- [KL96] KASKI S., LAGUS K.: Comparing self-organizing maps. In *Proceedings of International Conference on Artificial Neural Networks* (1996), pp. 809–814. [42](#)

- [KM08] KUIJPERS B., MOELANS B.: Towards a geometric interpretation of double-cross matrix-based similarity of polylines. In *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2008), pp. 1–8. [172](#)
- [KMB03] KARDOS J., MOORE A., BENWELL G.: The visualisation of uncertainty in spatially-referenced attribute data using trustworthy data structures. In *Proceedings of Annual Colloquium of the Spatial Information Research Centre* (2003), pp. 11–26. [32](#)
- [KMB05] KALNIS P., MAMOULIS N., BAKIRAS S.: On discovering moving clusters in spatio-temporal data. In *Proceedings of International Symposium on Spatial and Temporal Databases* (2005), pp. 364–381. [143](#)
- [KMS02] KEIM D., MÜLLER W., SCHUMANN H.: Visual data mining: State of the art report. In *Proceedings of Annual Conference of the European Association for Computer Graphics* (Sep 2002). [21](#), [24](#), [25](#)
- [KMS*08] KEIM D. A., MANSMANN F., SCHNEIDEWIND J., THOMAS J., ZIEGLER H.: Visual analytics: Scope and challenges, visual data mining: Theory, techniques and tools for visual analytics. *Lecture Notes In Computer Science (Incs) 4404* (2008), 76–90. [12](#), [13](#), [15](#), [19](#), [20](#), [39](#), [218](#)
- [KMSZ06] KEIM D. A., MANSMANN F., SCHNEIDEWIND J., ZIEGLER H.: Challenges in visual data analysis. In *Proceedings of International Conference on Information Visualisation* (2006), pp. 9–16. [21](#)
- [Kob04] KOBSA A.: User experiments with tree visualization systems. In *Proceedings of the IEEE Symposium on Information Visualization* (2004), pp. 9–16. [28](#)
- [Koh01] KOHONEN T.: *Self-Organizing Maps*, 3rd ed. Springer, Berlin, 2001. [41](#), [42](#), [89](#), [151](#), [176](#)
- [Koh05] KOHLHAMMER J.: *Knowledge Representation for Decision-Centered Visualization*. PhD thesis, TU Darmstadt, 2005. [219](#)
- [Koi94] KOIKKALAINEN P.: Progress with the tree-structured selforganizing map. In *Proceedings of European Conference on Artificial Intelligence. European Committee for Artificial Intelligence* (1994), pp. 211–215. [42](#)
- [Kra03] KRAAK M.: The space-time cube revisited from a geovisualization perspective. In *Proceedings of International Cartographic Conference* (2003). [143](#)
- [Kru56] KRUSKAL J. B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7, 1 (1956), 48–50. [49](#)
- [KSS06a] KEIM D. A., SCHNEIDEWIND J., SIPS M.: Scalable pixel based visual data exploration. *Lecture Notes In Computer Science 4370* (2006), 12–24. [43](#)
- [KSS06b] KLUKAS C., SCHREIBER F., SCHWÖBBERMEYER H.: Coordinated perspectives and enhanced force-directed layout for the analysis of network motifs. In *Proceedings of Asia-Pacific Symposium on Information Visualisation* (2006), pp. 39–48. [48](#), [55](#)
- [KTSZ08] KALBE T., TEKUŠOVÁ T., SCHRECK T., ZEILFELDER F.: GPU-accelerated 2D point cloud visualization using smooth splines for visual analytics applications. In *Proceedings of Spring Conference on Computer Graphics* (2008), pp. 111–125. [11](#), [28](#), [139](#), [144](#)
- [KV96] KRÁLOVIČ J., VLACHYNSKÝ K.: *Financial Management*. Elita, Bratislava, 1996. [192](#)
- [KW04] KEIM D., WARD M.: *Intelligent Data Analysis*. Springer Verlag, 2004, ch. Visual Data-Mining Techniques. [13](#), [16](#), [18](#), [21](#)

- [LAR99] LAWRENCE R. D., ALMASI G. S., RUSHMEIER H. E.: A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. *Data Mining and Knowledge Discovery* 3, 2 (1999), 171–195. 42
- [LCZ05] LIN L., CAO L., ZHANG C.: The fish-eye visualization of foreign currency exchange data streams. In *Proceedings of Asia-Pacific Symposium on Information Visualisation* (2005), pp. 91–96. 43
- [LF06] LESKOVEC J., FALOUTSOS C.: Sampling from large graphs. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006), pp. 631–636. 43, 50
- [LH07] LEE J., HAN J.: Trajectory clustering: A partition-and-group framework. In *Proceedings of ACM SIGMOD International Conference on Management of Data* (2007), pp. 593–604. 143
- [LJB06] LEGÁNY C., JUHÁSZ S., BABOS A.: Cluster validity measurement techniques. In *Proceedings of WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases* (2006), pp. 388–393. 42
- [LK07] LARAMEE R. S., KOSARA R.: Challenges and unsolved problems. *Lecture Notes in Computer Science* 4417 (2007), 231–254. 19, 219
- [LM07] LIU H., MOTODA H. (Eds.): *Computational Methods of Feature Selection*. Chapman and Hall, 2007. 40, 88
- [LPLS99] LA PORTA R., LOPEZ-DE-SILANTES F., SHLEIFER A.: Corporate ownership around the world. *The Journal of Finance* 54 (April 1999), 471–517. 102
- [LvW09] LI J., VAN WIJK J. J.: Evaluation of symbol contrast in scatterplots. In *IEEE Pacific Visualization Symposium* (2009), pp. 97–104. 222
- [MAN06] MAN AG: Man confirms interest in scania. http://www.man.eu/MAN-Downloadgalleries/EN/Press/PressRelease/2006/September_13_2006_Press_release_MAN_AG.pdf, September 2006. accessed 01.08.2009. 199
- [May07] MAY T.: Working with patterns in large multivariate datasets - Karnaugh-Veitch-Maps revisited. In *Proceedings of International Conference on Information Visualization* (2007), pp. 277–284. 27, 43
- [MCH*09] MOSCOVICH T., CHEVALIER F., HENRY N., PIETRIGA E., FEKETE J.-D.: Topology-aware navigation in large networks. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 2319–2328. 53
- [Mey98] MEYER B.: Self-organizing graphs - a neural network perspective of graph layout. In *Proceedings of International Symposium on Graph Drawing* (1998), pp. 246–262. 67
- [MFi] MFINDER AND MDRAW. <http://www.weizmann.ac.il/mcb/UriAlon/groupNetworkMotifSW.html>. accessed 05.06.2009. 54
- [MGT*03] MUNZNER T., GUIMBRETIERE F., TASIRAN S., ZHANG L., ZHOU Y.: TreeJuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. In *Proceedings of ACM SIGGRAPH* (2003), pp. 453–462. 55
- [MJ09] MCGUFFIN M. J., JURISICA I.: Interaction techniques for selecting and manipulating sub-graphs in network visualizations. *IEEE Transactions on Visualization and Computer Graphics* 15 (2009), 937–944. 53
- [MJW*09] MA'AYAN A., JENKINS S., WEBB R., BERGER S., PURUSHOTHAMAN S., HUSN N. A., POSNER J., FLORES T., IYENGAR R.: SNAVI: Desktop application for analysis and visual-

- ization of large-scale signaling networks. *BMC Systems Biology* 3, 10 (2009). 50, 54
- [MLM04] MICCICHE S., LILLO F., MANTEGNA R.: Correlation based hierarchical clustering in financial time series. In *Proceedings of Workshop of the international School of Solid State Physics "Complexity, metastability and nonextensivity"* (2004). 44
- [MM08] MUELDER C., MA K.-L.: Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov.–Dec. 2008), 1301–1308. 29, 52
- [MRH*05] MACEACHREN A. M., ROBINSON A., HOPPER S., GARDNER S., MURRAY R., GAHEGAN M., HETZLER E.: Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science* 32, 3 (July 2005), 139–160. 32
- [MSOI*02] MILO R., SHEN-ORR S., ITZKOVITZ S., KASHTAN N., CHKLOVSKII D., , ALONNETWORKMOTIFFSSCIENCE U.: Network motifs: Simple building blocks of complex networks. *Science* 298, 5594 (October 2002), 824–827. 71, 72
- [Mun97] MUNZNER T.: H3: laying out large directed graphs in 3D hyperbolic space. In *Proceedings of IEEE Symposium on Information Visualization* (1997), pp. 2–10. 29
- [Nan02] NANNI M.: *Clustering Methods for Spatio-Temporal Data*. PhD thesis, University of Pisa, 2002. 142, 174
- [NB02a] NESBITT K. V., BARRASS S.: Evaluation of a multimodal sonification and visualisation of depth of market stock data. In *Proceedings of International Conference on Auditory Display* (2002). 22
- [NB02b] NESBITT K. V., BARRASS S.: Evaluation of a multimodal sonification and visualisation of depth of market stock data. In *Proceedings of International Conference on Auditory Display* (2002). 22
- [NB05] NEUHAUS M., BUNKE H.: Self-organizing maps for learning the edit costs in graph matching. *Proceedings of IEEE Trans. on Systems, Man, and Cybernetics* 35 (June 2005), 503–514. 56
- [NCA06] NEUMANN P., CARPENDALE M. S. T., AGARAWALA A.: Phyllotrees: Phyllotactic patterns for tree layout. In *Proceedings of Eurographics IEEE VGTC Symposium on Visualization* (2006), pp. 59–66. 28
- [NFA01] NORTH C., FAROOQ U., AKHTER D.: Datawear: Revealing trends of dynamic data in visualizations. In *Proceedings of IEEE Symposium on Information Visualization* (2001), pp. 8–11. 143
- [NHM*07] NAM E. J., HAN Y., MÜLLER K., ZELENYUK A., IMRE D.: Clustersculptor: A visual analytics tool for high-dimensional data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (November 2007), pp. 75–82. 19, 42
- [Nor98] NORTH C.: A taxonomy of information visualization user-interfaces. <http://www.lirmm.fr/InfoViz/ASEval/References/shneiderman.php>, 1998. accessed on 1.9.2009. 22
- [NP06] NANNI M., PEDRESCHI D.: Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems* 27, 3 (2006), 267–289. 137, 142, 143, 145, 172, 174
- [NQ99] N.MATTHEWS, QIAN N.: Axis-of-motion affects direction discrimination, not speed discrimination. *Vision Research* 39, 13 (1999), 2205–2221. 157
- [OFS] O'MADADHAIN J., FISHER D., SMYTH P.: Analysis and visualization of network data using JUNG. *Journal of Statistical Software* VV, 2. 67, 91

- [OFW] O'MADADHAIN J., FISHER D., WHITE S.: Java universal graph framework (JUNG). 67
- [OKK04] ONNELA J., KASKI K., KERTESZ J.: Clustering and information in correlation based financial networks. *The European Physical Journal B* 38, 2 (March 2004), 353–362. 22
- [Oli99] OLIVE: Olive: On-line library of information visualization environments. <http://total.umd.edu/Olive/>, 1999. accessed on 1.9.2009. 22
- [OM02] OLSTON C., MACKINLAY J.: Visualizing data with bounded uncertainty. In *Proceedings of Information Visualization* (2002), pp. 37–40. 25
- [OPPROG09] OVALLE-PERANDONES M. A., PERIANES-RODRIGUEZ A., OLMEDA-GOMEZ C.: Hubs and authorities in a spanish co-authorship network. In *Proceedings of International Conference Information Visualisation* (2009), pp. 514–518. 54
- [Pö4] PÖLZLBAUER G.: Survey and comparison of quality measures for self-organizing maps. In *Proceedings of Workshop on Data Analysis* (2004), pp. 67–82. 42
- [PBB*08] PIKE W., BRUCE J., BADDELEY B., BEST D., FRANKLIN L., MAY R., RICE D., RIENSCHER R., YOUNKIN K.: The scalable reasoning system: Lightweight visualization for distributed analytics. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2008), pp. 131–138. 19
- [PBKA08a] PALMA A. T., BOGORNY V., KUIJPERS B., ALVARES L.: A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of ACM Symposium on Applied Computing, Advances in Spatial and Image-Based Information Systems Track* (2008), pp. 863–868. 137, 143
- [PBKA08b] PALMA A. T., BOGORNY V., KUIJPERS B., ALVARES L. O.: A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of ACM Symposium on Applied computing* (2008), pp. 863–868. 142
- [PD08] POHL M., DIEHL S.: What dynamic network metrics can tell us about developer roles. In *Proceedings of the international workshop on Cooperative and human aspects of software engineering* (2008), pp. 81–84. 54
- [PDR06] PÖLZLBAUER G., DITTENBACH M., RAUBER A.: Advanced visualization of self-organizing maps with vector fields. *Neural Networks* 19, 6 (2006), 911–922. 42
- [PGS09] PLAISANT C., GRINSTEIN G., SCHOLTZ J.: Visual-analytics evaluation. *IEEE Computer Graphics and Applications* 29, 3 (2009), 16–17. 219
- [PHP03] PFITZNER D., HOBBS V., POWERS D.: A unified taxonomic framework for information visualization. In *Proceedings of Asia-Pacific Symposium on Information Visualisation* (2003), pp. 57–66. 22, 24, 25
- [Pic95] PICHE S. W.: Trend visualization. In *Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering* (1995), pp. 146–150. 43
- [PKH04] PIRINGER H., KOSARA R., HAUSER H.: Interactive focus+context visualization with linked 2D/3D scatterplots. In *Proceedings of International Conference on Coordinated and Multiple Views in Exploratory Visualization* (2004), pp. 49–60. 23
- [PKJ*07] PU J., KALYANARAMAN Y., JAYANTI S., RAMANI K., PIZLO Z.: Navigation and discovery in 3D CAD repositories. *IEEE Computer Graphics and Applications* 27, 4 (2007), 38–47. 149
- [PKM*07] PELEKIS N., KOPANAKIS I., MARKETOS G., NTOUTSI I., ANDRIENKO G., THEODORIDIS Y.: Similarity search in trajectory databases. In *Proceedings of International Symposium on Temporal Representation and Reasoning* (2007), pp. 129–140. 137, 142, 172, 189

- [PM99] PAPADOPOULOS A. N., MANOLOPOULOS Y.: Structure-based similarity search with graph histograms. In *Proceedings of International Workshop on Database and Expert Systems Applications* (1999), pp. 174–178. 51, 87
- [Pow] POWERWALL SYSTEM, UNIVERSITY OF KONSTANZ, GERMANY.: PowerWall. <http://www.informatik.uni-konstanz.de/arbeitsgruppen/infovis/powerwall/>. accessed 01.12.2009. 184
- [PRD05] PÖLZLBAUER G., RAUBER A., DITTEN M.: Advanced visualization techniques for self-organizing maps with graph-based methods. *Lecture Notes in Computer Science* 3497 (2005), 75–80. 42
- [PSC05] PIROLLIN P., STUART CARD S. K.: The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis* (2005). 16
- [PSKN06] PANSE C., SIPS M., KEIM D., NORTH S.: Visualization of geo-spatial point sets via global shape transformation and local pixel placement. *IEEE Transactions on Visualization and Computer Graphics* 12 (September–October 2006). 149
- [PWL97] PANG A., WITTENBRINK C., LODHA. S.: Approaches to uncertainty visualization. *The Visual Computer* 13, 8 (Nov. 1997), 370–390. 32
- [RC94] RAO R., CARD S. K.: The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (April 1994), pp. 318–322. 28
- [Reu07] REUTERS: German DAX firms’ 2006 dividends seen rising 20 pct. <http://www.reuters.com/article/marketsNews/idUSL1436914220070314>, March 2007. accessed 01.08.2009. 196
- [RFF*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., TASKO J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov.–Dec. 2008), 1325–1332. 144, 222
- [RM06] RUDOLF MAYER THOMAS LIDY A. R.: The map of mozart. In *Proceedings of 7th International Conference on Music Information Retrieval* (October 2006), pp. 351–352. 42
- [RMC91] ROBERTSON G. G., MACKINLAY J. D., CARD S. K.: Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1991), ACM, pp. 189–194. 28
- [Rob03] ROBERTS P.: *Information Visualization for Stock Market Ticks: Toward a new trading interface*. Master’s thesis, MIT, December 2003. 23
- [Rob08] ROBINSON A.: Collaborative synthesis of visual analytic results. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 67–74. 19
- [RPN*08] RINZIVILLO S., PEDRESCHI D., NANNI M., GIANNOTTI F., ANDRIENKO N., ANDRIENKO G.: Visually-driven analysis of movement data by progressive clustering. *Information Visualization* 7 (2008), 225–239. 143, 145
- [RRAS08] ROYER L., REIMANN M., ANDREOPOULOS B., SCHROEDER M.: Unraveling protein networks with power graph analysis. *PLoS Computational Biology* 4, 7 (July 2008), e1000108+. 54
- [RW04] REN P., WATSON B.: Histograms: Interactive clustering of stacked graphs. In *Proceedings of IEEE Symposium on Information Visualization* (2004). 30

- [SBM08] SIMOFF S. J., BÖHLEN M. H., MAZEIKA A.: *Visual Data Mining*, vol. 4404. Springer, 2008, ch. Visual Data Mining: An Introduction and Overview, pp. 1–12. [12](#), [13](#), [16](#), [17](#), [21](#)
- [SBTK08] SCHRECK T., BERNARD J., TEKUŠOVÁ T., KOHLHAMMER J.: Visual cluster analysis in trajectory data using editable kohonen maps. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (Oct 2008), pp. 3–10. [vii](#), [7](#), [28](#), [139](#), [145](#)
- [SBvLK09] SCHRECK T., BERNARD J., VON LANDESBERGER T., KOHLHAMMER J.: Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization* 8, 1 (Spring 2009), 14–29. [vii](#), [viii](#), [7](#), [139](#)
- [SC00] SILVA S. F., CATARCI T.: Visualization of linear time-oriented data: A survey. In *Proceedings of International Conference on Web Information Systems Engineering* (2000), pp. 310–319. [143](#)
- [SCB*04] SCHMIDT G., CHEN S.-L., BRYDEN A., LIVINGSTON M., ROSENBLUM L., OSBORN B.: Multidimensional visual representations for underwater environmental uncertainty. *IEEE Computer Graphics and Applications* 24, 5 (Sept. 2004), 56–65. [32](#)
- [Sch02] SCHWERT W. G.: Anomalies and market efficiency. *Handbook of the Economics of Finance* 18 (2002), 937–972. [210](#)
- [Sch06] SCHOLTZ J.: Beyond usability: Evaluation aspects of visual analytic environments. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (Nov. 2006), pp. 145–150. [19](#)
- [Sch08] SCHWÖBBERMEYER H.: *Analysis of Biological Networks*. Wiley Series on Bioinformatics, Computational Techniques and Engineering. Wiley, 2008, ch. 5, pp. 85 – 112. [46](#), [51](#), [54](#), [72](#), [87](#)
- [SE04] SHEN X., EADES P.: Using MoneyTree to represent financial data. In *Proceedings of International Conference on Information Visualisation* (14–16 July 2004), pp. 285–289. [23](#)
- [SE05] SHEN X., EADES P.: Using moneycolor to represent financial data. In *Proceedings of Asia-Pacific Symposium on Information Visualisation* (2005), pp. 125–129. [23](#)
- [SH05] SMEULDERS R., HEIJS A.: Interactive visualization of high dimensional marketing data in the financial industry. In *Proceedings of the Ninth International Conference on Information Visualisation* (2005), pp. 814–817. [23](#)
- [SH06] SAWANT A. P., HEALEY C. G.: Visualizing abstract data using animation. In *Conference Compendium of IEEE Visualization 2006* (2006), pp. 37–38. [157](#)
- [Shn92] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-D space-filling approach. *ACM Transactions on Graphics* 11, 1 (1992), 92–99. [28](#)
- [Shn96] SHNEIDERMAN B.: The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages* (September 1996), pp. 336–343. [22](#), [23](#)
- [SKM06] SCHRECK T., KEIM D. A., MANSMANN S.: Regular treemap layouts for visual analysis of hierarchical data. In *Proceedings of Spring Conference on Computer Graphics* (2006). [31](#)
- [SLN05] SARAIYA P., LEE P., NORTH C.: Visualization of graphs with associated timeseries data. In *Proceedings of IEEE Symposium on Information Visualization* (Oct. 2005), pp. 225–232. [31](#)
- [SME08] SAVIKHIN A., MACIEJEWSKI R., EBERT D. S.: Applied visual analytics for economic decision-making. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2008), pp. 107–114. [43](#)

-
- [SMER06] SHEN Z., MA K.-L., ELIASSI-RAD T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Transactions on Visualization and Computer Graphics* 12, 6 (Nov.-Dec. 2006), 1427–1439. [19](#)
- [Smy97] SMYTH P.: Clustering sequences with hidden markov models. In *Advances in Neural Information Processing Systems* (1997), MIT Press, pp. 648–654. [142](#)
- [SP07] SCHRECK T., PANSE C.: A new metaphor for projection-based visual analysis and data exploration. In *Proceedings of IS&T/SPIE Conference on Visualization and Data Analysis* (2007), p. 64950L. [28](#), [144](#), [167](#)
- [Spe07] SPENCE R.: *Information Visualization - Design for Interaction*, 2nd ed. Pearson Education Limited, 2007. [15](#), [21](#), [33](#)
- [SS02] SEO J., SHNEIDERMAN B.: Interactively exploring hierarchical clustering results. *IEEE Computer* 35, 7 (2002), 80–86. [42](#)
- [SS04] SCHREIBER F., SCHWÖBBERMEYER H.: Towards motif detection in networks: Frequency concepts and flexible search. In *Proceedings of the International Workshop on Network Tools and Applications in Biology* (2004), pp. 91–102. [71](#), [72](#)
- [SS05] SCHREIBER F., SCHWÖBBERMEYER H.: Mavisto: a tool for the exploration of network motifs. *Bioinformatics* 21, 17 (July 2005), 3572–3574. [51](#), [54](#), [72](#)
- [SSH09] SCHULZ H.-J., SCHUMANN H., HADLAK S.: Point-based tree representation - a new approach for large hierarchies. In *Proceedings of IEEE Pacific Visualization Symposium* (2009), pp. 81–88. [28](#)
- [SSK07] SIPS M., SCHNEIDEWIND J., KEIM D. A.: Highlighting space-time pattern: Effective visual encodings for interactive decision making. *International Journal of Geographical Information Science* 21, 8 (2007), 879–894. [19](#)
- [SSZW08] SCHRECK T., SCHÜSSLER M., ZEILFELDER F., WORM K.: Butterfly plots for visual analysis of large point cloud data. In *Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (2008), pp. 33–40. [144](#), [167](#)
- [Sta00] STASKO J.: An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies* 53, 5 (2000), 663–694. [28](#)
- [STFK07] SCHRECK T., TEKUŠOVÁ T., FELLNER D., KOHLHAMMER J.: Trajectory-based visual analysis of large financial time series data. *SIGKDD Explorations* 9, 2 (December 2007), 30–37. [vii](#), [viii](#), [7](#), [11](#), [139](#), [143](#), [145](#)
- [STM*06] SAGHEER A., TSURUTA N., MAEDA S., TANIGUCHI R.-I., ARITA D.: Fast competition approach using self organizing map for lip-reading applications. In *Proceedings of International Joint Conference on Neural Networks* (2006), pp. 3775–3782. [42](#)
- [STT81] SUGIYAMA K., TAGAWA S., TODA M.: Methods for visual understanding of hierarchical system structures. *Systems, Man and Cybernetics, IEEE Transactions on* 11, 2 (Feb. 1981), 109–125. [52](#)
- [SWB02] SEKULER R., WATAMANIUK S. N. J., BLAKE R.: Perception of visual motion. In *Stevens' Handbook of Experimental Psychology* (2002), vol. 1, pp. 569–573. [158](#)
- [SZ00] STASKO J., ZHANG E.: Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Visualization* (2000), p. 57. [28](#)
-

- [Tar72] TARJAN R.: Depth first search and linear graph algorithms. *SIAM Journal on Computing* 1, 2 (1972), 146–160. [51](#)
- [TAS04] TOMINSKI C., ABELLO J., SCHUMANN H.: Axes-based visualizations with radial layouts. In *Proceedings of ACM symposium on Applied computing* (2004), pp. 1242–1247. [31](#)
- [TC05] THOMAS J. J., COOK K. A.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005. [12](#), [13](#), [19](#)
- [TC06] THOMAS J. J., COOK K. A.: A visual analytics agenda. *IEEE Computer Graphics and Applications* 26, 1 (Jan.-Feb. 2006), 10–13. [13](#), [15](#), [16](#), [19](#), [39](#), [218](#)
- [TG07] TESONE D., GOODALL J.: Balancing interactive data management of massive data with situational awareness through smart aggregation. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (2007), pp. 67–74. [19](#)
- [the] They rule. <http://www.theyrule.net/>. accessed 01.12.2009. [98](#)
- [THM*05] THOMSON J., HETZLER B., MACEACHREN A., GAHEGAN M., PAVEL M.: A typology for visualizing uncertainty. In *Proceedings of SPIE Conference on Visualization and Data Analysis 2005* (2005), vol. 5669, pp. 146–157. [25](#)
- [TK07] TEKUŠOVÁ T., KOHLHAMMER J.: Applying animation to the visual analysis of financial time-dependent data. In *Proceedings of International Conference on Information Visualization* (2007), pp. 101–108. [vi](#), [viii](#), [7](#), [11](#), [23](#), [30](#), [139](#), [143](#), [144](#)
- [TK08] TEKUŠOVÁ T., KOHLHAMMER J.: Visual analysis and exploration of complex corporate shareholder networks. In *Proceedings of the SPIE Visualization and Data Analysis* (jan 2008), vol. 6809, p. 68090F. [v](#), [vii](#), [4](#), [5](#), [29](#), [47](#), [54](#)
- [TKSK08] TEKUŠOVÁ T., KNUTH M., SCHRECK T., KOHLHAMMER J.: Data quality visualization for multivariate hierarchic data. In *Proceedings of International Symposium on Information Visualization* (2008), IEEE Computer Society, pp. 108–109. [11](#), [32](#)
- [TM02] TORY M., MÖLLER T.: *A Model-Based Visualization Taxonomy*. Tech. Rep. CMPT-TR2002-06, Computing Science Department, Simon Fraser University, 2002. [22](#)
- [TM04] TORY M., MOLLER T.: Rethinking visualization: A high-level taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization* (2004), pp. 151–158. [22](#)
- [TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies* 57, 4 (Oct. 2002), 247–262. [148](#), [155](#), [157](#)
- [TS07] TU Y., SHEN H.-W.: Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1286–1293. [31](#)
- [TS08] TEKUŠOVÁ T., SCHRECK T.: Visualizing time-dependent data in multivariate hierarchic plots - design and evaluation of an economic application. In *Proceedings of International Conference on Information Visualisation* (2008), pp. 143–150. [11](#), [31](#), [139](#)
- [TSB*08] TEKUŠOVÁ T., SCHRECK T., BEHR J., KOHLHAMMER J., STRICKER D.: The IGD-HEyeWall for visual analytics - concepts and applications. In *Proceedings of International Workshop on Giga-Pixel Displays & Visual Analytics* (2008). [139](#), [184](#)
- [TSPK08] TEKUŠOVÁ T., SKWAREK S., PARAMEI G., KOHLHAMMER J.: Perception of direction changes in animated data visualization. In *Symposium on Applied Perception in Graphics and Visualization* (2008), pp. 205–205. [vi](#), [7](#), [139](#)

- [TVK08] TEKUŠOVÁ T., VOSS V., KOHLHAMMER J.: Semantic search and visualization of time-series data. In *Proceedings of the 8th International Conference on Knowledge Management and International Conference on New Media Technology*. (2008), pp. 332–340. 148
- [Twe97] TWEEDIE L.: Characterizing interactive externalizations. In *Proceedings of the ACM CHI Human Factors in Computing Systems Conference* (March 1997), Pemberton S., (Ed.), pp. 375–382. 22
- [Ult95] ULTSCH A.: Self organizing neural networks perform different from statistical k-means clustering. Gesellschaft für Klassifikation, Basel, 1995. 42
- [Ult03] ULTSCH A.: *U*-matrix: a tool to visualize clusters in high dimensional data*. Tech. Rep. 36, Department of Mathematics and Computer Science, Philipps-University Marburg., December 2003. 42, 177
- [Uni09] UNILEVER N.V.: Annual report and accounts 2008 - adding vitality to life. <http://annualreport08.unilever.com/>, March 2009. accessed on 1.9.2010. 115
- [Ves99] VESANTO J.: SOM-based data visualization methods. *Intelligent Data Analysis* 3, 2 (1999), 111–126. 41, 42, 89, 177
- [VGK02] VLACHOS M., GUNOPOULOS D., KOLLIOS G.: Discovering similar multidimensional trajectories. In *Proceedings of International Conference on Data Engineering* (2002), pp. 673–684. 172
- [vHP09] VAN HAM F., PERER A.: Search, show context, expand on demand: Supporting large graph exploration with degree-of-interest. In *Proceedings of IEE Conference on Information Visualization* (2009). 53, 221
- [vHvW02] VAN HAM F., VAN WIJK J. J.: Beamtrees: compact visualization of large hierarchies. In *Proceedings of IEEE Symposium on Information Visualization* (2002), pp. 93–100. 28
- [vHW08] VAN HAM F., WATTENBERG M.: Centrality based visualization of small world graphs. *Computer Graphics Forum* 27, 3 (2008), 975–982. 50
- [vLBRS09] VON LANDESBERGER T., BREMM S., REZAEI P., SCHRECK T.: Visual analytics of time dependent 2D point clouds. In *Computer Graphics International* (2009), pp. 97–101. vi, viii, 7, 11, 31, 139, 163
- [vLGRS09] VON LANDESBERGER T., GÖRNER M., REHNER R., SCHRECK T.: A system for interactive visual analysis of large graphs using motifs in graph editing and aggregation. In *Vision Modeling Visualization Workshop* (2009), pp. 331–339. v, vii, 4, 5, 47
- [vLGS09] VON LANDESBERGER T., GÖRNER M., SCHRECK T.: Visual analysis of graphs with multiple connected components. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 155–162. vi, vii, 4, 5, 47
- [vLKS*10] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J. J., FEKETE J.-D., FELLNER D. W.: Visual analysis of large graphs. In *Proceedings of Annual Conference of the European Association for Computer Graphics* (2010), pp. 113–136. vii, 3, 11, 29, 47, 52, 53
- [VM04] VANDE MOERE A.: Time-varying data visualization using information flocking flocks. In *Proceedings of IEEE Symposium on Information Visualization* (2004), pp. 97–104. 157
- [VMMG04] VANDE MOERE A., MIUSSET K. H., GROSS M.: Visualizing abstract information using motion properties of data-driven infoticles. In *Proceedings of Conference on Visualization and Data Analysis - IS&T/SPIE Symposium on Electronic Imaging* (2004), pp. 33–44. 157

- [Š03] ŠIMUNIĆ K.: Visualization of stock market charts. In *Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (2003). [43](#)
- [vW05] VAN WIJK J. J.: The value of visualization. In *Proceedings of Visualization* (2005), pp. 79–86. [21](#)
- [WAG05] WILKINSON L., ANAND A., GROSSMAN R.: Graph-theoretic scagnostics. In *Proceedings of IEEE Symposium on Information Visualization* (2005), pp. 157–164. [145](#), [149](#), [163](#), [164](#), [165](#)
- [WAG06] WILKINSON L., ANAND A., GROSSMAN R.: High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics* 12, 6 (Nov.–Dec. 2006), 1363–1372. [145](#), [163](#), [164](#), [165](#), [166](#)
- [WAM01] WEBER M., ALEXA M., MÜLLER W.: Visualizing time-series on spirals. In *Proceedings of IEEE Symposium on Information Visualization* (2001), pp. 7–13. [30](#)
- [War00] WARE C.: *Information visualization: Perception for Design*. Morgan Kaufmann, 2000. [15](#), [21](#), [23](#), [25](#), [33](#), [34](#), [35](#), [36](#), [37](#), [53](#)
- [Wat06] WATTENBERG M.: Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), pp. 811–819. [54](#)
- [Wer06] WERNICKE S.: Efficient detection of network motifs. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 3, 4 (2006), 347–359. [51](#), [72](#)
- [Wiw01] WIWATTANAKANTANG Y.: *The equity ownership structure of Thai Firms*. Working Paper 2001-8, Hitotsubashi University, Center of Economic Institutions, 2001. [102](#)
- [WMC*09] WONG P. C., MACKEY P., COOK K., ROHRER R., FOOTE H., WHITING M.: A multi-level middle-out cross-zooming approach for large graph analytics. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 147–154. [53](#)
- [WR06] WERNICKE S., RASCHE F.: Fanmod: a tool for fast network motif detection. *Bioinformatics* 22, 9 (2006), 1152–1153. [51](#), [54](#), [72](#)
- [WS92] WATAMANIUK S. N. J., SEKULER R.: Temporal and spatial integration in dynamic random dot stimuli. *Vision Research* 32, 12 (1992), 2341–2347. [157](#)
- [WS03] WHITE S., SMYTH P.: Algorithms for estimating relative importance in networks. In *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), pp. 266–275. [54](#)
- [WvdWvW09] WILLEMS N., VAN DE WETERING H., VAN WIJK J. J.: Visualization of vessel movements. *Computer Graphics Forum* 28, 3 (2009), 959–966. [144](#)
- [YKSJ07] YI J. S., KANG Y. A., STASKO J., JACKO J.: Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1224–1231. [15](#), [33](#), [34](#), [35](#), [37](#)
- [ZK08] ZIEMKIEWICZ C., KOSARA R.: The shaping of information by visual metaphors. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov.–Dec. 2008), 1269–1276. [28](#)
- [ZMC05] ZHAO S., MCGUFFIN M. J., CHIGNELL M. H.: Elastic hierarchies: combining treemaps and node-link diagrams. In *Proceedings of IEEE Symposium on Information Visualization* (Oct. 2005), pp. 57–64. [29](#), [53](#)
- [ZNK07a] ZIEGLER H., NIETZSCHMANN T., KEIM D. A.: Relevance driven visualization of financial performance measures. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization* (2007), pp. 19–26. [43](#)

- [ZNK07b] ZIEGLER H., NIETZSCHMANN T., KEIM D. A.: Visual exploration and discovery of atypical behavior in financial time series data using two-dimensional colormaps. In *Proceedings of International Conference on Information Visualisation* (2007), pp. 308–315. [43](#)
- [ZWS96] ZHANG K., WANG J. T., SHASHA D.: On the editing distance between undirected acyclic graphs. *International Journal of Foundations of Computer Science* 7, 1 (1996), 43–57. [51](#), [87](#)